1-1-2016

Model-Based Analysis of User Behaviors in Medical Cyber-Physical Systems

Sanjian Chen
University of Pennsylvania, sanjian.chen3@gmail.com

Model-Based Analysis of User Behaviors in Medical Cyber-Physical Systems

Abstract
Human operators play a critical role in various Cyber-Physical System (CPS) domains, for example, transportation, smart living, robotics, and medicine. The rapid advancement of automation technology is driving a trend towards deep human-automation cooperation in many safety-critical applications, making it important to explicitly consider user behaviors throughout the system development cycle. While past research has generated extensive knowledge and techniques for analyzing human-automation interaction, in many emerging applications, it remains an open challenge to develop quantitative models of user behaviors that can be directly incorporated into the system-level analysis.

This dissertation describes methods for modeling different types of user behaviors in medical CPS and integrating the behavioral models into system analysis. We make three main contributions. First, we design a model-based analysis framework to evaluate, improve, and formally verify the robustness of generic (i.e., non-personalized) user behaviors that are typically driven by rule-based clinical protocols. We conceptualize a data-driven technique to predict safety-critical events at run-time in the presence of possible time-varying process disturbances. Second, we develop a methodology to systematically identify behavior variables and functional relationships in healthcare applications. We build personalized behavior models and analyze population-level behavioral patterns. Third, we propose a sequential decision filtering technique by leveraging a generic parameter-invariant test to validate behavior information that may be measured through unreliable channels, which is a practical challenge in many human-in-the-loop applications. A unique strength of this validation technique is that it achieves high inter-subject consistency despite uncertain parametric variances in the physiological processes, without needing any individual-level tuning. We validate the proposed approaches by applying them to several case studies.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Computer and Information Science

First Advisor
Insup Lee

Subject Categories
Computer Sciences

MODEL-BASED ANALYSIS OF USER BEHAVIORS IN MEDICAL CYBER-PHYSICAL SYSTEMS

Sanjian Chen

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2016

Supervisor of Dissertation

_____

Insup Lee

Professor, Computer and Information Science

Graduate Group Chairperson

_____

Lyle Ungar, Professor, Computer and Information Science

Dissertation Committee:

Oleg Sokolsky, Research Associate Professor, University of Pennsylvania

Rahul Mangharam, Associate Professor, University of Pennsylvania

James Won, Part-Time Lecturer, University of Pennsylvania

Stephen Patek, Associate Professor, University of Virginia

MODEL-BASED ANALYSIS OF USER BEHAVIORS IN MEDICAL CYBER-PHYSICAL SYSTEMS

Acknowledgements

First, I would like to thank and acknowledge my advisor, Dr. Insup Lee. It was only with his expert guidance, generous support, and unwavering encouragement that I was able to complete my doctoral research and this dissertation. He pushed me to think independently and constantly challenge myself to make me become a computer scientist. I consider myself very lucky to have the opportunity to work with him. I could not have asked for a better advisor.

I am extremely grateful to Dr. Oleg Sokolsky, the chair of my committee, for his insightful advice and confidence in me throughout my doctoral career. I would also like to thank the other members of my committee, Dr. Rahul Mangharam, Dr. James Won, and Dr. Stephen Patek, for providing critical feedback that helped significantly improve this dissertation.

The evaluation components of several case studies in this dissertation were only possible with the help of clinicians at the University of Pennsylvania Health System. I would like to thank and acknowledge all of our clinical collaborators, especially Dr. Michael Rickels, Dr. Benjamin Kohl, Amy Peleckis, Margaret Mullen-Fortino, and Dr. Soojin Park.

I was so fortunate to work with a group of great colleagues at the Penn PRECISE center. Special thanks to Dr. James Weimer and Dr. Lu Feng, whom I collaborated with on several research projects. During the past few years at Penn, I have learned a great deal from my fellow Ph.D. students and other members of the PRECISE center, who also became my good friends, including Andrew, Alex, David, Jian,

Paja, Nicola, Vincent, Baek-Gyu, Rado, Hao, Madhur, Matt, Katie, Peter, and many others. I would also like to thank several faculty members for giving advice to my work, especially Dr. Linh Thi Xuan Phan, Dr. George Pappas, Dr. Rajeev Alur, Dr. C. J. Taylor, and Dr. Chris Murphy.

Special thanks to Mike Felker, Liz, and the staff at the Moore Business Office. They handled all administrative tasks professionally and perfectly.

My research was supported in part by several grants, including NSF CNS-1035715. I completed the Ph.D. program with full fellowship support from the University of Pennsylvania. I am very grateful to all the funding sources.

Finally, no word can express my deepest gratitude to my parents and my girlfriend. I spent very limited time with them in the past few years. They had to overcome great emotional difficulties. Their unconditional love and support have encouraged me to complete the Ph.D. degree. I hope this achievement could make them proud and bring them some comfort.

ABSTRACT

MODEL-BASED ANALYSIS OF USER BEHAVIORS IN MEDICAL CYBER-PHYSICAL SYSTEMS

Sanjian Chen

Insup Lee

Human operators play a critical role in various Cyber-Physical System (CPS) domains, for example, transportation, smart living, robotics, and medicine. The rapid advancement of automation technology is driving a trend towards deep human-automation cooperation in many safety-critical applications, making it important to explicitly consider user behaviors throughout the system development cycle. While past research has generated extensive knowledge and techniques for analyzing human-automation interaction, in many emerging applications, it remains an open challenge to develop quantitative models of user behaviors that can be directly incorporated into the system-level analysis.

This dissertation describes methods for modeling different types of user behaviors in medical CPS and integrating the behavioral models into system analysis. We make three main contributions. First, we design a model-based analysis framework to evaluate, improve, and formally verify the robustness of generic (i.e., non-personalized) user behaviors that are typically driven by rule-based clinical protocols. We conceptualize a data-driven technique to predict safety-critical events at run-time in the presence of possible time-varying process disturbances. Second, we develop a methodology to systematically identify behavior variables and functional relationships in healthcare applications. We build personalized behavior models and analyze population-level behavioral patterns. Third, we propose a sequential decision filtering technique by leveraging a generic parameter-invariant test to validate behavior information that may be measured through unreliable channels, which is a practical challenge in many human-in-the-loop applications. A unique strength of

this validation technique is that it achieves high inter-subject consistency despite uncertain parametric variances in the physiological processes, without needing any individual-level tuning. We validate the proposed approaches by applying them to several case studies.

Contents

List of Tables

List of Illustrations

Chapter 1

Introduction

1.1 Motivation

Human operators have a critical role in many Cyber-Physical System (CPS) application domains, e.g., transportation [208, 121, 202], smart living [185, 284], robotics [80, 68, 71], and medicine [169]. In some cases, such as self-driving cars [26], fully autonomous control with near-zero human intervention may be achievable but still requires extensive research efforts before provably safe and effective products are available to the general public. In many domains including health care, entertainment, and avionics, humans play an essential role that is unlikely to be completely replaced by automation in the foreseeable future. The past decade has observed significant progress in promoting the level of automation in a broad range of CPS systems: examples include advanced driver assistance features [41], flight control systems [282], unmanned aerial/ground vehicles [184], robotic surgeries [204], and smart infusion pumps [215]. Driven by the advancement of autonomy, a trend towards deeper "human-automation teamwork" starts to emerge in many areas [280, 159, 131].

The presence of human behaviors in the operation loop introduces new challenges to system design and analysis. Compared to software-based controllers, humans exhibit very different characteristics in information processing, decision making, and

execution [273]. Unlike how computing systems work, human judgment and actions are shaped by complex physiological, behavioral, and psychological factors [206, 273]. While automation can eliminate some types of human errors by substituting operators in performing certain tasks, it also introduces new human factor challenges, e.g., the well-noted "mode confusion" problems in pilot-automation interaction [241, 240, 275]. Human factor research has shown that automation can cause unanticipated changes to how humans perform cognitive tasks [219]. Some human-automation issues have led to tragic accidents: For example, divergence between the pilots' mental model and the actual autopilot behavior is believed to be a contributing factor to the 2013 Asiana 214 San Francisco crash [186, 251]; the 2016 Tesla fatal crash, in which the autopilot system was activated but suffered a detection failure, highlights the challenge to ensure safety of self-driving cars that still require driver supervision [195].

Ample evidence suggests that the traditional technology-centered design approach is insufficient in coping with new challenges introduced by the increasingly sophisticated interaction between human behaviors and technology [273]. Designing highly reliable Human-in-the-Loop (HiL) CPS requires a new holistic engineering paradigm that explicitly consider behaviors in the design process [245]. We need new methodologies to systematically analyze the implications of operator behaviors on system-level properties.

1.2 Why Model-Based Analysis is Useful

Safety-critical HiL systems ultimately need to be evaluated in live tests before actual deployment. For example, automotive manufacturers conduct extensive road tests before releasing new vehicles, and most, if not all, life-critical medical devices must pass human clinical trials to obtain regulatory approval. The main limitation is that testing life-critical systems in humans usually involves significant risks and costs.

For example, due to the safety concerns, some critical medical devices must pass "preclinical" tests [135] to prove that they are reasonably safe before they are allowed to be evaluated in human trials. As HiL CPS become increasingly more complex, evaluating different design choices by human tests becomes more costly and risky. Additionally, it is impossible to cover all operation scenarios solely by testing.

Model-based analysis is a particularly useful methodology to complement testing in the development cycle. For medical systems, model-based analysis may substitute certain costly preclinical trials, especially during the earlier design stages. The rationale is that model-based analysis can efficiently rule out improper designs in a risk-free manner, which not only enables quick system prototyping but also saves cost and possibly also lives in subsequent human trials. One recent successful example is the Type 1 Diabetes Metabolism Simulator (T1DMS) [75], which is accepted by the U.S. Food and Drug Administration (FDA) as the first software tool that can be used to substitute animal tests in certain pre-clinical trials of glucose control algorithms. In addition to evaluating automation design, model-based analysis results can also provide feedback to refine user behaviors. Model-based simulators are used to train operators, e.g., surgical simulators [243] and driving simulators [94].

1.3 Research Problems and Challenges

Scope of this dissertation. A lot of research has been done in applying model-based analysis to CPS applications that consist of only non-human components, e.g., mechanical and electrical systems [165, 232, 79]. This dissertation proposes new modeling paradigms and techniques for analyzing user behaviors in the emerging generation of HiL medical CPS that encompass complex interactions between humans, physiology, and technology. These so-called "sociotechnical" systems recently started to garner increasing attention from the CPS community [206]. More specifically, we focus on modeling user behaviors and their impact on the safety of

medical CPS that include physiological processes, sensors, actuators, and human & non-human control agents. Although we focus on medical applications, the proposed methodologies can be applied to other HiL CPS domains.

Applying model-based analysis to HiL CPS involves two key research problems: How to model behaviors and how to use behavior models in analysis.

Modeling behaviors can be broken down into three sub-problems: Identifying relevant behaviors, applying proper modeling techniques, and validating models. Determining which behaviors should be modeled involves a deliberate trade-off of a few factors: Relevance (does the behavior has a major impact on the relevant system properties?), observability (can the behavior be measured with reasonable cost?), modeling difficulty (can the behavior be quantified and modeled?), and model utility (will the model be useful for analysis?).

One of the main purposes of modeling behaviors is to generate quantitative insights into how human factors impact the operational properties of the entire HiL systems. The behavior model and models of other system components need to be expressed at the same level so that meaningful closed-loop analysis (e.g., simulation and formal verification) can be done. If analysis results reveal that system properties may be violated, then the user behaviors and/or the design of non-human components must be refined.

There are several challenging issues that are especially relevant to model-based analysis of human behaviors in medical CPS:

Identifying quantifiable behavior metrics. Modeling requires precisely defined quantifiable metrics. In many HiL applications, engineers are interested in understanding certain behavioral patterns, e.g., aggressive driving [139, 199] and automation trust [219]. Those patterns are higher-level information that provides deep insights into behavioral traits. The challenging issue is how to quantify descriptive behavioral trends such as "aggressiveness" and "trust" in the application context.

Non-determinism of behaviors. Unlike automation agents, humans are influenced by physiological (e.g., level of attentiveness) and psychological factors (e.g., trust and mood) [273] that their behaviors can be highly personalized and inherently probabilistic. Such non-determinism introduces challenges to analysis.

Uncertainties in physiological processes. In medical CPS, patients' physiological parameters can vary across different individuals and the same patient's parameters can exhibit short-term fluctuations in certain scenarios. Constrained by available sensing technology, many physiological parameters are simply not measurable [187]. As a result, the same action may trigger drastically different observable physiological responses, making it challenging to analyze the physiological impact of behaviors.

Unreliable behavior measurements. Behavior measurements are sometimes collected through unreliable channels. For example, some smart infusion pumps rely on possibly erroneous patient self-reported information to calculate recommended doses [247], and some driver-assistance systems use computer vision techniques to infer driver poses [248], which have inherent misdetection rates. Faulty behavior information may jeopardize safety: For example, if a smart insulin infusion pump receives incorrect eating information, it may deliver insulin unnecessarily and impose life-threatening hypoglycemia risk.

1.4 Overview of the Proposed Approach

This dissertation focuses on developing methods to solve the behavior modeling problems and address the challenges that are discussed in the previous section. We start with a key observation that the user behaviors in many medical CPS can be broadly categorized into one of two types: Generic behaviors and personalized behaviors. Generic behaviors are seen in systems in which users are expected to exhibit

pre-defined behaviors that are designed to work in a target patient population, i.e., the behaviors are not tuned to each individual on-the-fly. This type of behaviors is common in hospital care: Standard clinical protocols guide how caregivers conduct treatments and interact with medical devices. For example, clinicians use pain management protocols to control opioid consumption for post-surgery patients [228]. The protocols define a set of rules that apply to all patients within the target population. On the other hand, personalized behaviors are shaped by users' individual discretions and preferences, which are common in out-patient healthcare applications. For example, some Type 1 diabetics use smart insulin pumps to help control blood glucose [2]. The users' meal intake and exercise habits can significantly impact their glucose levels as well as the pump's operation, and those behaviors are highly personalized.

We model generic behaviors primarily based on domain knowledge, e.g., the protocols that drive the user behaviors. For personalized behaviors, we develop a data-driven technique to individualize the behavior model. To address the challenge that some behavior measurements might be unreliable, we design a novel technique to validate the behavior information considering uncertain individual physiological parameters.

After developing the behavior models, we integrate them with models of other components, such as automation and physiological processes, to analyze the safety of the closed-loop systems. To this end, we propose two analysis paradigms to fit the different requirements for generic and personalized behaviors. Generic behaviors must maintain safety over the entire target population, i.e., they must be "robust" against the possible inter-subject physiological variances. We harness the strengths of numerical simulation and hybrid system verification tools to achieve a synergy in ensuring robustness of generic behaviors. Personalized behaviors are expected to be "adaptive" to individual physiology and are inherently statistical, i.e., the daily carb intake of an insulin pump user is most likely a random variable with a certain

distribution rather than a fixed value. We leverage machine learning techniques to analyze the relevant behavior trends from data and use probabilistic model checking to verify the individualized physiological impacts of personalized behaviors. The closed-loop analysis results provide feedback on how behaviors, either generic or personalized, can be revised to ensure safety.

1.5 Summary of Contributions

This dissertation makes the following contributions:

1. We propose a model-based analysis framework to evaluate, improve, and verify the robustness of generic (i.e., non-personalized) behaviors driven by rule-based protocols. We apply the framework to an intraoperative glycemic control case study: we identify the weaknesses of a current protocol, design an enhancement, and formally verify the new protocol using a state-of-the-art physiological model. We verify that the new protocol maintains safety over a virtual patient population that maps to continuous ranges of uncertain physiological states and parameters. Our verification work provides a new level of safety guarantee than other simulation-based evaluation methods that can only cover finite discrete samples of the virtual population. Additionally, we develop a novel virtual-subject based, data-driven run-time safety monitor technique to predictively alert caregivers to critical events in the presence of possible time-varying, unobservable physiological disturbance.

2. We develop a "Time-Apps-Physiology triggered Living-Treatment actions" (TAP-LT) framework to systematically identify behavior metrics and functions in patient-centered healthcare applications. We design a data-driven method to instantiate the TAP-LT framework to represent personalized behaviors. We apply the methodology to an insulin pump case study and identify quantifiable user behavior patterns. The analysis results reveal new clinical insights

that enable more efficient and personalized diagnosis (confirmed by expert clinical review). We formally evaluate the individualized physiological impacts of switching behavior patterns by probabilistic model checking. The verification results suggest patients may improve clinical outcomes by behavioral change.

3. We design a model-based detection method to validate unreliable real-time behavioral measurements. The detector leverages a generic parameter-invariant test that is enhanced by a sequential decision filtering technique. An important feature of the proposed validation technique is high inter-subject performance consistency despite physiological variances across different individuals. We apply the technique to a diabetic meal detection case study and design a novel meal detector. Simulation and clinical evaluations demonstrate that our meal detector provides the highest detection rate, lowest false alarm rate, and shortest detection time, compared to three other state-of-the-art meal detectors. Moreover, our detector achieves consistent inter-subject performance without any individual-level parameter tuning.

1.6 Outline of the Dissertation

The rest of this dissertation is organized as follows:

Chapter 2 reviews related work including the Human-Automation Interaction (HAI) research and a few recent CPS projects on using behavior models for system-level analysis. We discuss how this dissertation work will complement existing body of knowledge.

Chapter 3 introduces a model-based analysis framework to evaluate, improve, and verify the robustness of protocol-guided generic behaviors in medical systems. We apply the framework to a concrete clinical case study, in which we enhance an existing clinical protocol to overcome its weaknesses and formally prove that the new protocol is safe over a virtual population. To address the challenge of unobservable

time-varying physiological disturbance, we introduce a data-driven safety monitor technique to predict critical events at run time.

Chapter 4 describes the TAP-LT framework to systematically identify personalized behavior metrics and a data-driven technique of instantiating quantifiable personalized behavior features. We apply the proposed approach to an insulin pump case study and analyze individualized diabetic user behaviors.

Chapter 5 proposes a behavior event validation method that is designed to achieve a consistent detection performance despite parametric variances across individuals. We apply the technique to a meal detection case study and show that the new detector significantly outperforms other state-of-the-art detectors.

Chapter 6 concludes the dissertation and discusses future research opportunities.

Chapter 2

Background

Researchers in varies fields of engineering, psychology, and computer science have studied issues stemmed from human-automation interactions (HAI) since the 80s [36, 125, 136, 273]. Modeling behaviors for system-level analysis of HiL systems becomes an inter-disciplinary area that started to garner increasing attention in the CPS community over the past few years. Research in this domain has so far been done in a bottom-up fashion, i.e., modeling problems are formulated in the application context, and the techniques are tailored to specific case studies. In this chapter, we first survey some of the key results of the HAI research in Section 2.1. In Section 2.2, we review a few recent CPS projects on using behavior modelings for system-level analysis. Section 2.3 concludes this chapter and discusses how this dissertation contributes to the existing body of research.

2.1 Human-Automation Interaction

The engineering community has long recognized that automation can create new problems, sometimes more than it eliminates, when it interacts with human operators [17]. Human-automation interaction (HAI) related problems have been identified in the investigations of a number of catastrophic system failures in areas

including transportation [257, 96], power plants [214], and medicine [148]. The accidents have stimulated over two decades' active research by human factor and control engineering communities in understanding HAI issues and designing better human-automation interfaces [120, 175, 233]. Despite numerous results and design improvements in HAI engineering, it remains a grand challenge to systematically identify potentially risky interactions between human operators, automation, and the physical processes being controlled [226].

In this section, we review some of the key results and insights of past HAI research that are closely related to this dissertation. Section 2.1.1 describes the HAI issues identified by engineering psychology research. Section 2.1.2 surveys research by computer scientists and system engineers on applying formal methods to modeling and verifying HAI.

2.1.1 Engineering Psychology Research Related to HAI

Engineering psychology researchers approach the HAI problem mainly from the perspective of considering how automation can change the nature of human cognitive functions and subsequently lead to accidents. In the widely-cited textbook [273], Wickens et al. identify several automation and human performance issues, among which uncalibrated automation "trust" and inappropriate automation state feedback are highlighted [273].

One of the most-studied HAI issue is "trust" [17, 205, 275, 171]. Human operators may over-trust or under-trust automation [218], and both can compromise safety. Over-trust arises when operators believe automation is highly reliable and develop the tendency of not paying enough attention to monitoring system functions when they should. When automation failures do occur, distracted humans may be less capable of handling emergencies quickly and properly. Engineers use the term "complacency" to describe such phenomenon [276]. Another form of over-trust, closely related to complacency, is automation bias [203]. It describes over-trusting users as-

signing more authority to automation than they should, and thereby increasing the likelihood of following incorrect guidance when automation malfunctions. In addition to causing loss of situation awareness, over-trust can also lead to "deskilling" [172]. For example, reliance on auto-pilot systems may result in degradation of manual fly skills among pilots [239]. Under-trust describes the phenomenon that operators ignore automation outputs even when they are actually correct, because automation is perceived to be unreliable or too complex to apprehend [219]. One example is the widely recognized "alarm fatigue" problem in healthcare [72]. Getting tired of too many false alarms [82], caregivers may totally ignore all alarms including the true ones.

Another issue that has been extensively studied by human factor engineers is feedback to humans on automation states [210]. A lot of research has been done in aviation systems concerning the interaction between pilots and autopilots. "Mode confusion" describes the phenomenon that pilots may misjudge which mode the autopilot is functioning in, jeopardizing safety in critical situations [240].

Recognizing the various HAI issues and their adverse impact on safety and system performance, researchers advocate the need of a design paradigm shift from technology-centered perspective to human-centered automation [29]. Several factors have been highlighted in promoting the safety and efficacy of HAI, including designing appropriate levels of automation [274], efficient automation feedback [246], and calibrating automation trust through training [16].

2.1.2 Formal Analysis of HAI

HAI has also attracted attention from the formal methods community. Bolton, Bass, and Siminiceanu write a review article that surveys more than 100 publications related to formally verifying HAI [36]. As pointed out in their review, research in this area broadly falls into two categories: Those that concern the interface between humans and automation, and those that include the whole system into the modeling

scope [36].

Human-automation interfaces (HCI) have been formally modeled and checked against desired properties. Most existing work model HCI as some forms of state transition systems [220, 105, 81, 86, 35]. The properties being checked are typically expressed using temporal logic [59] and mainly concern usability [37]. Campos and Harrison identify four classes of HAI properties that can be formally checked: Reachability [223], visibility [48], task related [1], and reliability [49]. A major research branch of formally verifying HCI is dedicated to identifying potential mode confusions [174], for which several techniques have been proposed to apply model checking to verifying interfaces [45] and analyzing human mental/knowledge models [211, 21, 127].

Another research thrust in formally verifying HAI expands the scope of modeling from focusing on the interface to considering system-level properties. In their review article [36], Bolton, Bass, and Siminiceanu categorize research along this direction into two types: Those that consider measurable behaviors, and those that aim at understanding the cognitive factors that drive observable behaviors. The former class focuses on using formal models to represent tasks [143]. Several formalisms have been proposed, including operator function models [201, 37], user action notation [108], and concur-tasktrees [224]. The later class concentrate on modeling cognitive behaviors and operators' knowledge in interacting with automation. A number of cognitive modeling methods have been developed, including the operator choice model [51, 179], programmable user model [46, 32], and distributed cognition models [192].

2.2 Model Operator Behaviors in CPS

In the problem space of modeling human-automation teamwork in CPS, research progress has been made recently in several applications. The techniques are typically

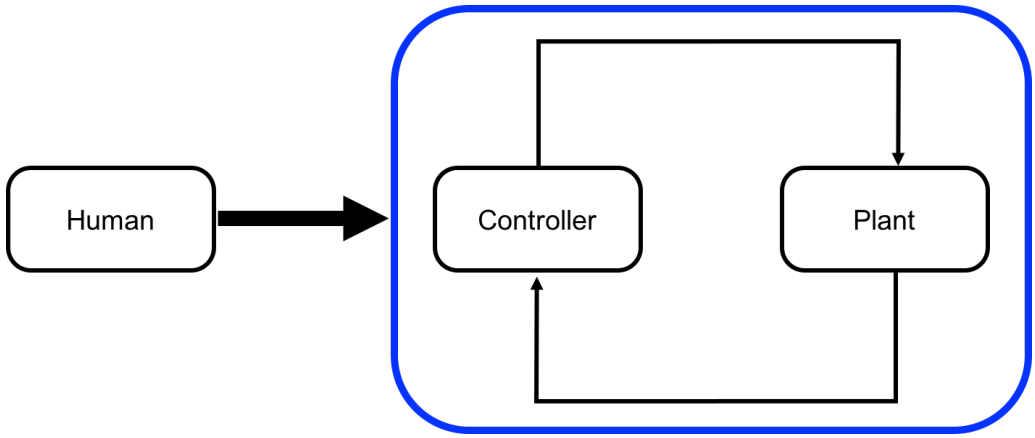


Figure 2.1: The open-loop view of a behavior model.

configured and evaluated in the application context. The specific characteristics of the applications make them amenable to the corresponding modeling methods. In this section, we review three recent projects by other research groups, including semi-autonomous driving [248], human-UAV teamwork modeling [5], and living-assistance robot [265]. Those projects represent three different types of behavior models (open-loop model, closed-loop model, and system-level model) and cover a range of model analysis techniques (simulation, formal verification, and statistical analysis). In the rest of this section, we briefly review the case studies and discuss their strengths and limitations.

2.2.1 Open-Loop Behavior Model

The open-loop behavior model treats behaviors as a disturbance to the coupled system of the automatic controller and the controlled plant, as illustrated in Figure 2.1. Webster et al. apply the open-loop behavior modeling to a living assistant robot application [265, 266, 255, 254].

Problem description. They conduct experiments using a commercially available Care-O-bot "robot butler" [234]. The robot can locate itself and navigate in a house that is equipped with various sensors. The robot can receive and interpret high-level

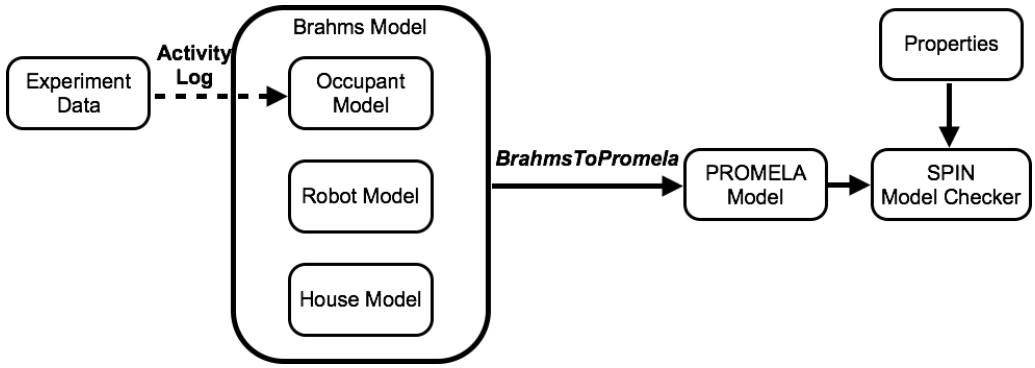


Figure 2.2: Overview of the modeling method in the human-care robot case study [265].

operation commands through its robot operating system, e.g., "put an object on the tray and move to the living room". The human-robot teamwork system includes three components: The human, robot, and environment. The system model describes the operation logic and interaction relations between the components. Their approach solves the following formal verification problem: Given a model of the system and a set of requirements, prove or disprove that all possible executions of the system model satisfy the requirements.

Methodology overview. Figure 2.2 shows their overall methodology. They model the human-robot system in a multi-agent modeling and simulation language called Brahms [250], for which formal operational semantics have been defined [255]. The specifications are expressed in linear temporal logic, which is a formalism that enables specifying temporal properties, e.g., "at all points in time something must be true" (the global `G` operator).

The activity data is recorded from a person living in the robot house for four days [85]. The user activity log is used to construct three different modeling scenarios: A deterministic scenario that is directly converted from the activity log; a nondeterministic scenario that allows random selection of any of the 26 unique activities identified from the activity log; a nondeterministic conjoined activity scenario that extends the nondeterministic scenario and decomposes overlapping events

into mutually exclusive events by renaming. The three scenario models are verified against four sample requirements, which are expressed as formalized properties in linear temporal logic (LTL), e.g., "the robot will remind the person that medicine is due at 5 PM every day.". The four properties are verified on all three models using the SPIN model checker. All properties are satisfied by the three models.

Remarks. Their approach demonstrates the utility of formal methods in verifying functional properties of the human-robot application. The Brahms language allows modeling the interaction of agents using `IF-THEN` style statements. The *BrahmsToPromela* tool automatically translates Brahms models into PROMELA models that the SPIN model checker accepts.

For safety verification, the formal model must exactly represent or over-approximate the target system with respect to the requirements: The possible model execution traces must form a superset of real system behaviors; if the formal model does not express all possible behaviors of the real system, passing model checking does not guarantee that the real system always satisfy the requirements. In the robotic assistant application, the nondeterministic scenario models assume the human agent can arbitrarily take any action at a given time step, which is an over-approximation of reality. In some applications where the system is very well conditioned, over-approximation can easily lead to requirement violations, in which case no conclusion can be made about whether the real system would violate the requirements. In those situations, identifying a non-trivial over-approximated model to generate meaningful verification results can be a major challenge. Another limitation is that the properties in the presented work [265] are a few disjointed simple examples, and they do not seem to constitute a complete, coherent set of system requirements.

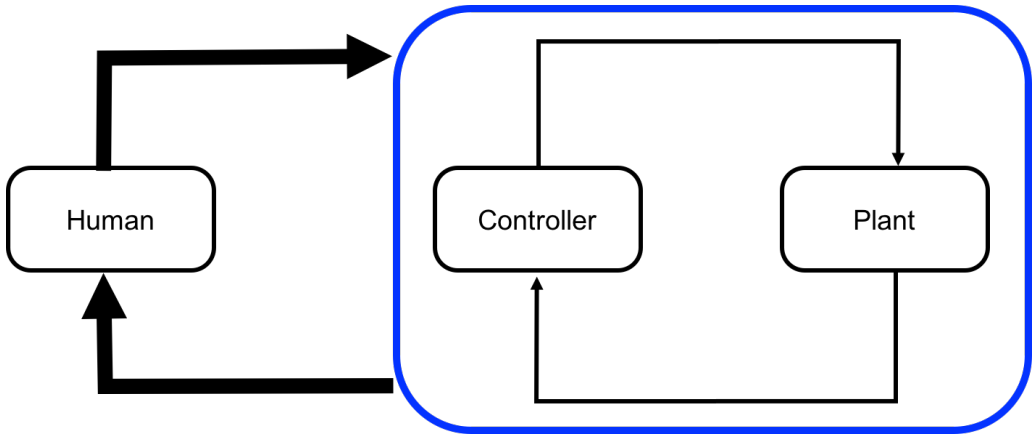


Figure 2.3: The closed-loop view of a behavior model.

2.2.2 Closed-Loop Behavior Model

The closed-loop behavior model explicitly considers information feedback to operators, as shown in Figure 2.3. Shia et al. apply the closed-loop behavior model to a semi-autonomous driving application [248, 238, 230, 263, 84].

Problem description. Figure 2.4 illustrates the workflow of semi-autonomous driving. The driver is given a specific driving task, e.g., make turns or go straight. The driver's actions are influenced by both the driver's state, e.g., distracted or attentive, and the environmental conditions, e.g., the presence of obstacles. The semi-autonomous controller has three key components. The first component predicts future vehicle trajectories given the driver's inputs, e.g., steering, acceleration, and braking, under different environmental conditions and driver states. The second component compares the predicted vehicle trajectories with unsafe regions, e.g., an obstacle or road curbs, and decides whether the controller needs to apply a correction to the driver's inputs. If the controller decides to intervene, the third component computes the control inputs, e.g., steering angle.

The model-based analysis framework must solve three technical problems that correspond to the three components of the semi-autonomous controller shown in Figure 2.4:

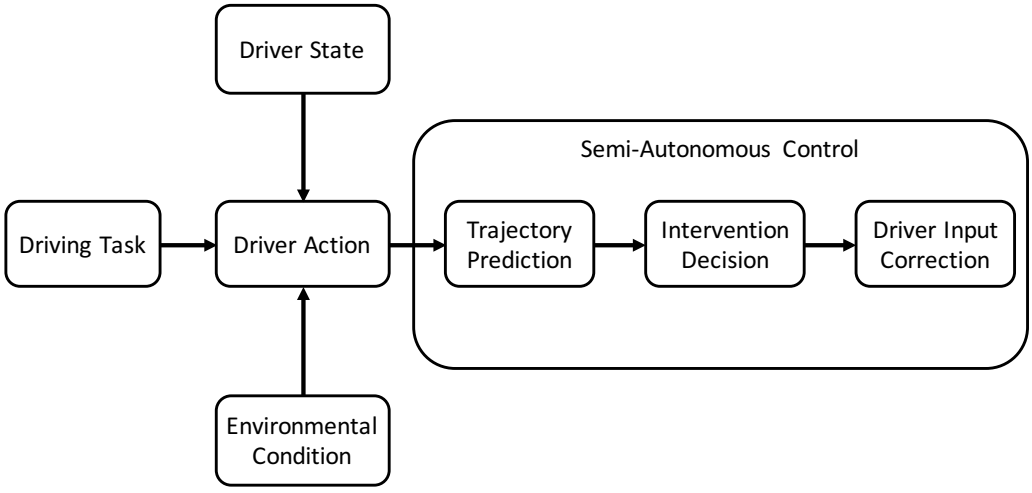


Figure 2.4: The workflow of semi-autonomous driving.

1. At each time point, given the measurements of the driver's pose in a past time window and the information of the vehicle as well as its environment in a future time window, predict the driver's action and the resulting vehicle trajectory within a near-future time window.

2. Given the vehicle trajectory predictions and the environmental conditions, decide if the vehicle could enter unsafe regions.

3. Given the driver's input, the vehicle information, and the environmental condition, compute correctional control inputs, if necessary, such that the vehicle stays in the safe region.

Methodology overview. Twenty-four drivers participate in a two-hour driving session on the industry driving simulator CarSim. The first hour is for driver data collection, during which the semi-autonomous controller is turned off, and the second hour is for testing, during which the controller is activated. The drivers need to perform specific driving tasks, e.g., maintain a certain speed and/or keep a safe distance from the leading car. Drivers may be distracted by text messages. The

simulator emulates obstacles by sudden speed drops of the leading car or appearances of simulated animals, forcing the driver to take defensive maneuvers. The full combinations of environmental conditions (with or without obstacles) and the driver's states (with or without phone distraction) yield four possible driving scenarios. A Microsoft Kinect monitors the driver pose in real time, and it tracks the 3-D movements of the driver's joints using computer vision technologies. The simulator records vehicle dynamics measurements, the driver's inputs, and road information.

Shia et al. propose a procedure that uses the training data to learn the mapping from the driver pose and environmental conditions to the driver's actions. They associate the driver pose data with the environmental condition at each time step, and apply k-means algorithm [107] to cluster the combined dataset. For each cluster, they identify the driver's actions in the next 1.2 seconds time window and pass the inputs into a vehicle model [231] to predict trajectories.

The expected vehicle trajectories are intersected with unsafe regions that are defined by obstacle locations, the lane boundaries, and the road boundaries. The autonomous controller intervenes if the intersection is non-empty [248], and the control inputs are calculated using the standard Model Predictive Control (MPC) technique [47].

The driver-controller performance is evaluated in the second hour of the simulated driving experiment. The key result is that at a medium clustering setting, the semi-autonomous controller intervenes 93% of the instances when the driver is in danger; 71% of times that the controller chooses to intervene, the driver is going to be in danger in a near future time window. The semi-autonomous controller keeps the vehicle safe during the entire testing period for all drivers.

Remarks. Their technique enables validating the safety of a semi-autonomous driving system by incorporating a driver behavior model. The proposed framework is modular, i.e., the driver and controller are explicitly modeled as separate compo-

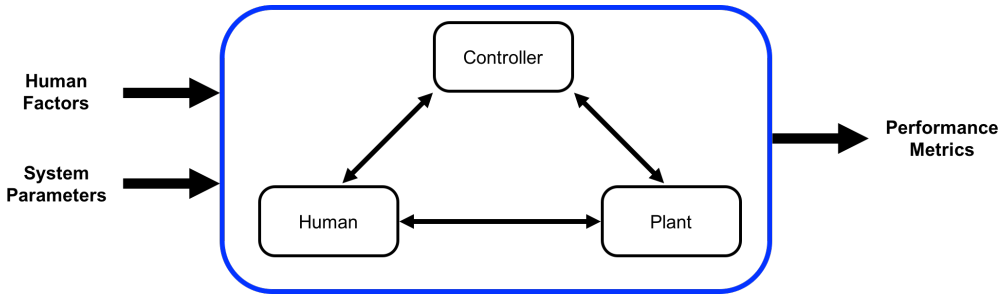


Figure 2.5: The system-level view of a behavior model.

nents, and thus it allows the use of more complex driver and/or control models.

One limitation is that the semi-autonomous controller is designed and tested on exactly the same group of drivers. As a result, any possible selection bias is unlikely to be revealed in the testing, i.e., the semi-autonomous controller may fail to handle certain situations that are not observed on those recruited into the study. Additionally, the approach does not account for the "behavioral feedback", i.e., drivers may behave differently when they know there is a semi-autonomous controller acting as a safety backup. In the experiments of the surveyed work, the drivers know whether the controller is activated, and many drivers admit that the controller changes their own behaviors [248]. This is a critical issue in HAI: The automation may impact human psychology and make them more willing to engage in aggressive control actions. The problem is also noted by researchers in other HiL application domains, e.g., the closed-loop medical devices [222].

2.2.3 System-Level Behavior Model

The system-level behavior model, shown in Figure 2.5, focuses on a holistic analysis of the input-output performance. Ahmed et al. develop statistical models that predict the system-level performance of human-UAV networks [5, 6, 4, 38].

Problem description. Thirty volunteers participate in a simulated study using the Aptima's Dynamic Distributed Decision-making simulation software. With the

assistant of system messages, each participant controls eight friendly UAVs to accomplish simulated tasks with varying task load (TL) and message quality (MQ). The participants are evaluated using a version of the standard Operation Span (OSPAN) test to measure their Working Memory (WM) capacity [90]. Four performance metrics were evaluated for each subject during the experiment: Red Zone Performance (RZP), Time to Destroy enemy Target (DT), and Enemy Destroyed Performance (EDP). Ahmed et al.'s work' solves the following problem: Given TL, MQ, and WM, develop a statistical model to predict the four performance metrics, RZP, DT, EDP, and AE.

Methodology overview. Ahmed et al. apply three statistical models to solving the prediction problem [5]. As a starting point, classic linear regression is applied to the experimental data [31]. To address the limitations of linear regression (LR), two alternative models are evaluated: Gaussian processes (GP) directly relate predictions to the training data, and Bayesian networks (BN) enable inverse inference.

Results. The LR, GP, and BN models are evaluated using cross validation. All models achieve good prediction performance (an error rate of 3.63%) on EDP. Closer examination of the raw data reveals that a lot of false predictions are borderline cases, where either the predicted value is close to the discretization cut-offs or has a high variance. Confidence thresholds are introduced to reject such borderline cases, i.e., the model does not make a prediction if the MAP value probability does not exceed the corresponding confidence threshold. Introducing the confidence thresholds significantly improves the prediction performance of all models on the three performance metrics.

Remarks. Their work applies several statistical models to predicting the performance of the human-UAV network. It reveals several insights into the advantages and limitations of each model with respect to the application. LR has an easy-

to-understand form but may generate unreliable predictions when the testing point is far from any sampled point, which may be problematic when the space is high-dimensional and can only be sparsely sampled due to experimental constraints. In the GP model, the predicted uncertainty range adapts to the similarity between the testing point and sampled points. However, tuning a GP model requires carefully picking the kernel function and non-convex optimization of parameters [31]. BN explicitly captures the conditional probabilistic dependencies between variables, and it enables inverse inference. But learning the BN structure can be challenging and often requires heuristics to trim down the model selection space. Moreover, the variable discretization has a significant impact on the model complexity and prediction performance.

The technique is an offline, black-box modeling approach: It concerns only system-level characteristics and high-level performance metrics; the model does not capture any runtime operation dynamics. The approach is applicable when it is reasonable to conjecture that system characteristics that are measured offline (e.g., task load, message quality, and operator cognitive factors) can sufficiently explain all performance metrics. If a system has highly complex internal dynamics, the relationship between the system characteristics and performance is likely to be highly complex, non-linear, and harder to model. Moreover, some system characteristics may change over time, e.g., task load, and the offline model does not account for temporal changes in parameters. The model prediction results can guide high-level system configurations, but they are probably insufficient to inform low-level real-time control and/or decision support at the operational level, because the model does not capture any operational details.

2.3 Discussion

Designing safe and effective HAI in complex systems requires developing a deep holistic understanding of various aspects including human factors, control engineering, user interface design, and risk analysis. Plenty of research opportunities exist in working towards an integrated analytical framework that allows engineers to build, validate, and use behavior models to uncover useful insights for system design.

The engineering psychology research lay out basic understanding of what types of issues need to be considered in designing HAI. The insights from psychology studies and accident investigations generate (often qualitative) guidelines and principals that can be useful for developing quantitative models for rigorous system analysis. For example, one of the most important issues that needs to be considered in analyzing most HAI systems is automation trust.

Previous research on applying formal methods to analyzing HAI have made notable progress towards modeling and verification [34, 20], but how to develop individualized behavior model and use it in closed-loop analysis with well-conditioned physical processes remains a largely open problem. HCI research limit the scope to the interface and do not explicitly include other components of the system in the models. Most existing work on formally modeling the entire system either concerns mechanical/electrical system whose dynamics are well-understood [19, 217, 37], or use abstract environmental models to hide the detailed dynamics of the controlled process [34].

This dissertation complements existing work on the model-based analysis of HAI in several aspects. First, our proposed approach considers the end-to-end design problem from developing behavior model to using behavior model in closed-loop analysis and generating feedback to improve behaviors and/or automation design. Second, we evaluate behavior models by coupling them with models of well-conditioned physiological processes. Third, we propose a method that allows the automation agent to validate potentially unreliable information on operator behaviors. Previous

research on state feedback in HAI mostly concerns the opposite direction of information flow: Feedback from automation to humans, where information reliability may not be a prominent issue as it is machine recorded. As we will describe in Chapter 5, the information channel from humans to automation can be unreliable and may compromise safety if automation receives false behavior measurements.

Chapter 3

Model-Based Analysis of Generic Behaviors

This chapter considers medical CPS in which user behaviors are "generic", i.e., the behaviors are not adapting on-the-fly to physiological variances among different patients. This type of behavior is common in many in-hospital medical systems, where caregivers follow established protocols to interact with medical devices in treatment procedures [58, 163]. The protocols standardize medical practice for treating a certain patient population. Therefore, it is crucial to validate that a protocol is "robust", i.e., it is safe for everyone within the target population despite possible individual physiological differences.

Hospitals currently design protocols mostly based on literature survey and medical consensus among a local group of physicians [69]. Most notably, clinicians typically tune the protocol towards the anticipated "average" physiological response based on their clinical experiences. The problem is that the protocol may not be safe on those "outlier" subjects who are extremely sensitive or insensitive to treatments. Furthermore, it is risky to test different protocol designs by repeated experiments on humans, especially considering that many protocols apply to surgical or critically-ill patients. There is a critical need for a method to validate the robustness of a protocol

design without incurring significant life risk.

In this chapter, we propose a model-based analysis framework to evaluate, improve, and verify the robustness of protocol-guided user behaviors in medical CPS. By applying the framework to a clinical case study, we are able to formally prove that an enhanced surgical insulin protocol is safe over a virtual patient population that maps to continuous regions of uncertain physiological parameters and unmeasurable initial physiological states. To the best of our knowledge, this is the first attempt towards formally verifying an insulin protocol using the most advanced glucose/insulin metabolism model, which contains unidentifiable parameters and unobservable states. To address the practical challenge that a patient's physiological parameters may change over time, we propose a novel data-driven computational virtual subject based adaptive technique for ensuring run-time safety using the most advanced physiological model.

Part of the work described in this chapter has been published in our previous papers [147, 55].[1]

The rest of this chapter is organized as follows: Section 3.1 motivates and formulates the problem of modeling and evaluating generic behaviors driven by rule-based protocols; Section 3.2 describes our approach; Section 3.3 presents a case study in which we apply the model-based analysis technique to identifying the weaknesses of an existing protocol, designing an enhancement, formally verifying the safety of the new protocol design, and ensuring run-time safety in the presence of temporal physiological variances; Section 3.4 concludes our work in this research thrust

3.1 Problem Description

In current hospital care, clinicians are responsible for taking measurements from sensing devices (e.g., vital sign monitors) and changing configurations on therapy

[1]The publishers and/or the copyright agreements grant using any portion of the papers in a dissertation.

devices (e.g., infusion pumps). Clinical protocols are standardized procedures that guide medical practice [58, 163]. Examples include insulin infusion protocols for glucose level regulation [258], analgesia protocols for pain management [228], sedation control protocols [69], and ventilator weaning protocols [89]. Clinicians are expected to follow those protocols, although deviations can happen due to practical reasons, e.g., nurses may delay or miss a protocol-specified check point because of an emergency situation.

Designing a protocol to reliably achieve a clinical goal, particularly when faced with patient-specific physiologic parameters such as insulin sensitivity is, at best, challenging and, at worst, harmful [162, 52, 93]. Current clinical protocols are mostly derived from experience and intuition. They are typically developed by consensus among local groups of clinicians, often taking into account available resources from the medical literature review. It is common that different institutions use their own protocols for the same clinical scenario [69]. However, it is unclear which protocol design would result in better clinical practice and outcomes because it is neither feasible nor ethical to repeatedly test all potential variations of a protocol on human patients.

By consulting healthcare practitioners at the Hospital of the University of Pennsylvania and reviewing a number of current clinical protocols for different medical scenarios including insulin infusion, sedation control, and pain management, we find that most protocols share a similar rule-based logical structure: They define a set of clinical metrics to be monitored (e.g., vital signs) and variables that can be controlled (e.g., infusion rates); the main logic is specified as a set of rules that tell clinicians when to measure the monitoring variables and how to set control variables accordingly. As an example, a rule in an insulin infusion protocol may be "if the glucose reading has dropped by 30 mg/dL or less from last measurement (30 minutes ago), then decrease insulin rate by 2 U/h and do not give any insulin bolus". This chapter concerns user behaviors that are driven by those rule-based clinical protocols, which

we formally define as follows:

Definition 1 *A rule-based protocol is a tuple* $\langle w, \mathbf{y}, \mathbf{u}, L \rangle$ *that consists of four components:*

- ***Monitoring variables****, denoted as* $\mathbf{y} \in \mathcal{R}_{\mathbf{y}}$*, which is a vector of physiological variables that need to be monitored, defined in the space of* $\mathcal{R}_{\mathbf{y}}$*.*
- ***Control variables****, denoted as* $\mathbf{u} \in \mathcal{R}_{\mathbf{u}}$*, which is a vector of control variables that can be set, defined in the space of* $\mathcal{R}_{\mathbf{u}}$*.* [2]
- ***A set of*** n ***rules,*** *denoted as* $L := \bigcup_{i=1}^{n} \{M_i\}$*, where* M_i *is the i-th rule specified as* $M_i : G_i(\mathbf{y}) == True \rightarrow \mathbf{u} = \mathbf{u}^i$*.* $G_i(\mathbf{y})$ *is a function that maps* $\mathbf{y}$ *to a boolean value.* M_i *dictates that if* $G_i(\mathbf{y})$ *is True, then assign the predefined value* $\mathbf{u}^i$ *to* $\mathbf{u}$*. In addition,* L *must be* ***consistent*** *and* ***complete****. Consistency requires that* $\forall \mathbf{y} \in \mathcal{R}_{\mathbf{y}},\ \forall i$ *and* $j \in \{1, \ldots, n\}, \mathbf{G}_i(\mathbf{y}) \wedge \mathbf{G}_j(\mathbf{y}) \neq True$*, i.e., no more than one rule can be enabled by a measurement* $\mathbf{y}$*. Completeness requires that* $\forall \mathbf{y} \in \mathcal{R}_{\mathbf{y}}, G_1(\mathbf{y}) \vee \ldots \vee G_n(\mathbf{y}) = True$*, i.e., a measurement* $\mathbf{y}$ *enables at least one rule.*
- ***Sampling period,*** *denoted as* w*. At the beginning of each period,* $\mathbf{y}$ *is sampled and* $\mathbf{u}$ *is updated according to a rule in* L*.*

The research problem is to evaluate the safety and robustness of behaviors driven by the rule-based protocols. In the clinical environment, safety objectives are commonly defined in terms of keeping the relevant physiological variables within the target regions. Because a rule-based protocol is supposed to work on the entire target patient population, it must be robust, i.e., it should maintain safety for everyone within the target population, regardless of possible inter-subject physiological variances.

[2] Here we follow the control system notation convention, where $\mathbf{y}$ represents system measurements and $\mathbf{u}$ represents control variables.

3.2 A Model-Based Analysis Framework

In this section, we propose an analytical framework that leverages physiological models to ensure safety and robustness of behaviors driven by rule-based protocols. Closed-loop analysis requires precise semantics of the execution of rule-based protocols. Therefore, we start by modeling protocol-driven behaviors as hybrid systems. The hybrid system behavior model is first evaluated in numerical simulation against the physiological model with discrete samples of physiological parameters and initial states. Numerical simulation allows efficient protocol prototyping and is particularly useful at ruling out unsafe protocol designs. However, because simulation can only cover finite samples of the physiological parameters and initial states, passing simulation evaluation does not guarantee safety on everyone within the target population in all possible scenarios. To ensure robust safety, we further evaluate candidate protocol designs in formal verification, which enables exhaustively checking the entire state space given ranges of physiological parameters and initial states. Our framework achieves synergy between numerical simulation and formal verification: The former enables efficient testing and revision of protocol design but does not provide hard safety guarantees with respect to uncertain physiological parameters and states; the latter provides robust safety guarantees but is computationally more expensive.

The rest of this section is organized as follows: Section 3.2.1 presents hybrid system modeling of protocol-driven behaviors; Section 3.2.2 reviews physiological modeling and the key challenges that are related to using physiological models; Section 3.2.3 discusses how we integrate numerical simulation and formal verification to ensure robustness of protocol-driven behaviors, focusing on addressing the challenges associated with using physiological models.

3.2.1 Model Protocol-Driven Generic Behaviors

A hybrid system is a formal model of systems that include both discrete and continuous dynamics [111]. It provides a convenient formalism to model many CPS that include both discrete behaviors of digital components and continuous dynamics of physical systems. Modeling, control, and verification of hybrid systems have been an active research field in the past two decades [166, 79, 142].

A hybrid system is an automaton with discrete states and continuous variables. The discrete states form a graph structure, in which transitions between the states are triggered by conditions defined over the continuous variables. Each discrete state is associated with a "flow" function, which is an equation of the continuous variables and their first derivatives: While the hybrid system is at a certain discrete state, the continuous variables evolve along a differential curve as defined by the flow function. A formal definition of a hybrid system is given as follows [112, 10]:

Definition 2 *A hybrid system is a tuple* $\mathcal{H} = \langle \mathcal{X}, \mathcal{Q}, \mathcal{X}_{init}, \mathcal{X}_{inv}, \mathcal{F}, T \rangle$*:*

- $\mathcal{X}$ *represents the vector of continuous variables.*
- $\mathcal{Q}$ *denotes the vector of discrete states.*
- $\mathcal{X}_{init} \in \mathcal{R}_{\mathcal{X}}$ *specifies an initial condition to each discrete state.* $\mathcal{H}$ *may start from a discrete state whose initial condition is true.*
- $\mathcal{F}$ *assigns a flow function to each discrete state. A flow function is a predicate over* $\mathcal{X}$ *and its first derivative* $\dot{\mathcal{X}}$*. While* $\mathcal{H}$ *stays at a discrete state,* $\mathcal{X}$ *evolves along the differential curve defined by the flow function.*
- $\mathcal{X}_{inv}$ *maps each discrete state to an invariant condition. An invariant condition is a predicate over* $\mathcal{X}$ *that must remain true while* $\mathcal{H}$ *stays at the discrete state.*
- T *maps each of the transitions between* $\mathcal{Q}$ *to a guard condition. A guard condition is a predicate over* $\mathcal{X} \bigcup \mathcal{X}'$*, where* $\mathcal{X}'$ *is the updated value of* $\mathcal{X}$ *after a transition is taken.*

By Definition 1, a protocol is driven by external measurements sampled at a certain frequency, i.e., the protocol is a reactive system that needs to be constantly driven by external inputs. However, note that the coupled system of a protocol and the corresponding physiological process forms a closed loop that can be modeled as a hybrid system, i.e., the physiology is driven by control inputs from the protocol, and as the physiological system evolves, it feeds updated physiological measurements to the protocol. This section focuses on the part of the hybrid system that describes protocol-driven behaviors. In the following description of the hybrid system model, we use an abstract function $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$ to represent the physiological process that specifies how outputs $\mathbf{y}$ evolve given inputs $\mathbf{u}$.

Note that the actual user behaviors may deviate from the protocol rules. This may be caused by many practical limitations in healthcare, e.g., clinicians may not be able to take measurements at exactly the check points defined by the protocol because they are busy handling emergencies, something that is not uncommon in operating rooms and intensive care units. To capture such practical deviations, our behavior model expresses the check point timing variations, i.e., the actual update action may happen within an uncertain time window around the specified check point.

Next, we present the hybrid system model of the closed-loop system that consists of behaviors driven by a rule-based protocol $\langle w, \mathbf{y}, \mathbf{u}, L \rangle$ and the physiological process. The basic idea is to create a hybrid system with one discrete state that would periodically take self-transitions to update $\mathbf{u}$ according to the protocol. Each protocol rule is mapped to one self-transition. We use a continuous variable t to model time passage, which gets reset to zero at each self-transition and flows at a constant rate of 1 in the discrete state. When the automaton stays at the discrete, $\mathbf{y}$ evolves according to the physiological process function $\mathbf{f}(\mathbf{y}, \mathbf{u})$, and $\mathbf{u}$ is only updated on the self-transitions. Here is a formal definition of our hybrid system model of a rule-based protocol driven behaviors.

Definition 3 *Given a rule-based protocol* $\langle w, \mathbf{y}, \mathbf{u}, L\rangle$ *and a physiological process represented as a function* $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$*, the hybrid system model of the closed-loop system is* $\mathcal{H} = \langle \mathcal{X}, \mathcal{Q}, \mathcal{X}_{init}, \mathcal{X}_{inv}, \mathcal{F}, T\rangle$*:*

- $\mathcal{X} = \{t, \mathbf{y}, \mathbf{u}\}$*, where* t *is the time variable.* $\mathbf{y}$ *and* $\mathbf{u}$ *are defined by the protocol.*
- $\mathcal{Q} = \{Q_0\}$ *is a single discrete state denoted as* Q_0*.*
- $\mathcal{X}_{init}(Q_0) = True$*, i.e., the system starts at* Q_0*.*
- $\mathcal{F}(Q_0) = \{\dot{t} = 1, \dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}), \dot{\mathbf{u}} = 0\}$*, representing that at* Q_0*, the time variable* t *progresses at a constant rate,* $\mathbf{y}$ *progresses according to the physiological process function* $\mathbf{f}(\mathbf{y}, \mathbf{u})$*, and control inputs* $\mathbf{u}$ *stay constant because* $\mathbf{u}$ *is only updated on the transitions, when the protocol is executed.*
- $\mathcal{X}_{inv}(Q_0) = \langle t \leq w + \delta\rangle$*, where* w *is the protocol's sampling period and* δ *denotes that the actual measurement time may deviate from the specified check point by at most* $\pm\delta$*.*
- $T = \bigcup_{i=1}^{n}\{T_i\}$*, where* T_i *denotes the self-transition that corresponds to the* i*-th rule:* $T_i = (t \geq w - \delta \wedge G_i(\mathbf{y}) = True \xrightarrow{Q_0 \to Q_0} \mathbf{u} = \mathbf{u}^i \wedge t = 0)$*. The* $t \geq w - \delta$ *condition, together with* $t \leq w + \delta$ *in the invariant* $\mathcal{X}_{inv}(Q_0)$ *would force one of the self-transitions to happen within the time interval* $[w-\delta, w+\delta]$*. The* $Q_0 \to Q_0$ *denotes that all transitions are self-transitions from* Q_0 *to* Q_0*. The* $G_i(\mathbf{y}) = True$ *condition selects the self-transition that corresponds to the rule being activated.* T *basically expresses that within the time interval* $t \in [w - \delta, w + \delta]$*, the self-transition that corresponds to the activated rule will be taken, and after the transition,* $\mathbf{u}$ *will be updated according to the enabled rule and* t *will be reset to zero.*

Note that the physiological process may add additional continuous and discrete states to the model of the closed-loop system. The hybrid system presented here only

highlights the behaviors part of the system and uses an abstract function $\mathbf{f}(\mathbf{y}, \mathbf{u})$ to represent the entire physiological process. In Section 3.3, we present a case study of a real closed-loop system in which the physiological process itself also contains both continuous and discrete dynamics.

3.2.2 Physiological Models

Clinicians and bio-medical engineers have been studying the problem of developing mathematical models of physiological processes for decades. For example, textbooks by Cobelli and Carson [50, 62] are good accounts of this subject. Latest advances in bio-medical engineering have led to high fidelity models of certain physiological systems. A recent successful example is that the United States Food and Drug Administration (FDA) has accepted the Type 1 Diabetes Metabolic Simulator (T1DMS) as the first software tool that can be used to substitute animal test in pre-clinical trials of glucose control algorithms [151]. With physiological models, it is possible to analyze the clinical outcomes of protocol-driven behaviors in risk-free model-based evaluation.

The most commonly used technique in describing first-principle physiological processes is "compartmental modeling". Physiological phenomena typically involve distribution and interaction of substances (e.g., a medication or hormone) between different parts of the body, which can be described by compartmental models [50, 65, 62, 64, 124, 100]. A compartmental model is typically a set of differential/difference equations in which the state variables represent the quantities of the target substances within each compartment and the equations represent the flows of the substances between the compartments. The rates at which substances enter or leave compartments are represented by model parameters. Since most compartmental models are derived from first-principle physiology, the parameters usually have actual physiological meanings: For example, a parameter may represent how fast insulin moves from tissue to blood.

There are several key challenges in using physiological models to evaluate operator behaviors by closed-loop analysis:

1. **Unidentifiable physiological parameters:** The parameters in compartmental models represent the generation, transportation, and bio-chemical interaction rates of substances at certain parts of the body, and in many cases, the parameters are not identifiable by standard medical devices. For example, one of the parameters in an advanced glucose/insulin model [187] represents the insulin exchange rates between liver and plasma, which are not measurable in current hospital settings.

2. **Unobservable initial physiological states:** The state variables in compartmental models represent the quantities of substances. Some body compartments are hard-to-reach by current sensing technologies, and the corresponding states are therefore not measurable. One example from the glucose/insulin model is the total mass of insulin in interstitial fluids, which cannot be directly measured by available sensors. A major implication for analysis is that the initial states of physiological models may be partially unknown. For example, a surgical glucose model [55] contains seven states, out of which only one (i.e., the plasma glucose concentration) is measurable during surgeries. How the unmeasurable states are initialized can have a profound impact on the analysis results: For example, the glucose trajectories of two model instances may turn out completely different when both start at the same observable glucose level but differ in the unobservable initial quantities of insulin in the body.

3. **Non-linear dynamics:** Some physiological models contain non-linear terms that represent complex interactions between physiological states. The non-linearity may significantly increase the difficulty of analysis and formal verification.

3.2.3 Closed-Loop Safety Analysis

To ensure robustness of protocol-driven behaviors, we integrate hybrid system behavior models with physiological models in closed-loop analysis and verify the safety requirements, e.g., whether the relevant physiological metrics are kept within the safe zone. To address the aforementioned challenges associated with using physiological models, we design an analysis process that leverages clinical knowledge about physiological parameters and states. Although some physiological parameters and states can not be feasibly measured on every patient in the hospital, their ranges of values and statistical distributions over a particular patient population may be derived from clinical studies [106, 76, 155, 138, 158]. The information on the distributions of parameters enables generating "virtual subjects", each of which is an instantiation of model parameters [75]. A physiological model, together with its virtual subjects, constitute a testbed for "virtual" clinical trials (also called *in silico* trials in some literature) to evaluate treatment methods.

Model-based virtual clinical trials have been used to test therapy strategies in a number of medical applications, e.g., cardiac pacemakers [129], glucose control [63], and pain management [14]. The analysis techniques can be broadly categorized into two classes: numerical simulation and formal verification.

Numerical simulation is widely used in model-based testing of clinical procedures [183, 277]. Many simulation tools exist, e.g., Matlab/Simulink. The main limitation is that simulation can only sample finite points in the space of parameters and initial states. Therefore, passing simulation evaluation does not guarantee safety over the entire population that maps to the continuous regions in the space of physiological parameters and initial states.

Formal verification is the process of checking whether a system satisfies given properties. Model checking is an automatic formal verification technique [60]. Figure 3.1 shows the general workflow of model checking. The system is described in a formal modeling language, e.g., PROMELA [115] or timed automata [11]. The

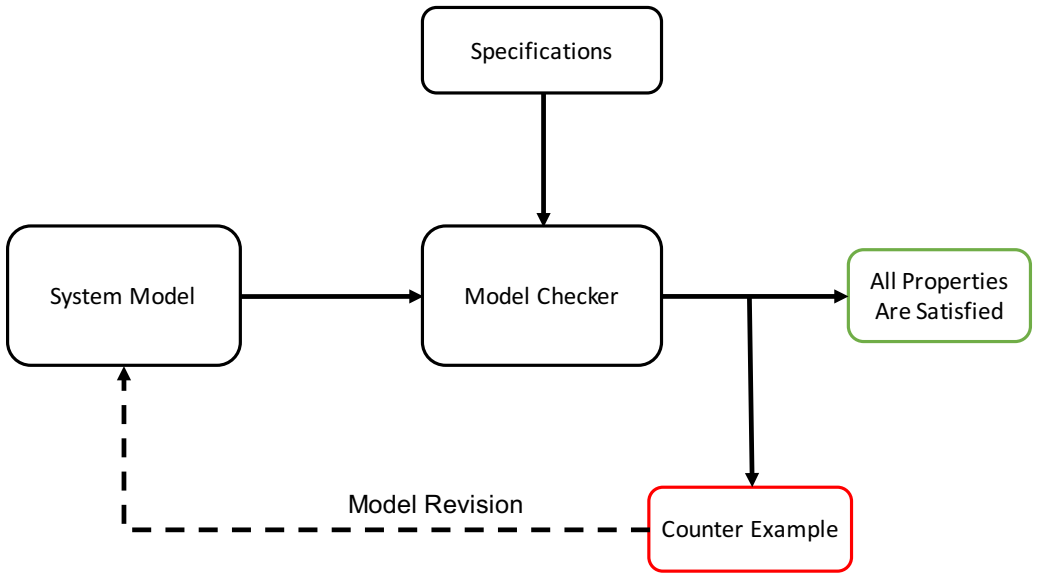


Figure 3.1: The general workflow of model checking.

specifications include a set of properties written in logical formulas, e.g., the Linear Temporal Logic (LTL) [262]. A model checker, e.g., SPIN [115] or UPPAAL [160], exhaustively checks the properties along all possible execution paths of the system model, and it either generates a counterexample with a particular execution path that violates some of the properties or reports that all properties are satisfied. The counter example can guide revisions of the system design. A practical challenge in model checking stems from the state space explosion problem: The size of the state space grows exponentially with the size of the system [196]. Numerous techniques have been proposed to tackle this challenge [9, 141].

Model checking has emerged as a powerful technique with successful applications to many practical problems [24] and has recently been applied to medical applications [130, 129, 14, 57]. A significant portion of existing medical device verification work focuses on verifying cardiac pacemakers [130, 129, 57], where physiological models are represented as linear timing functions with observable states. Arney et al. propose a technique to simulate and verify patient controlled analgesia algorithms using a linear, observable physiological model. A key design assumption in their

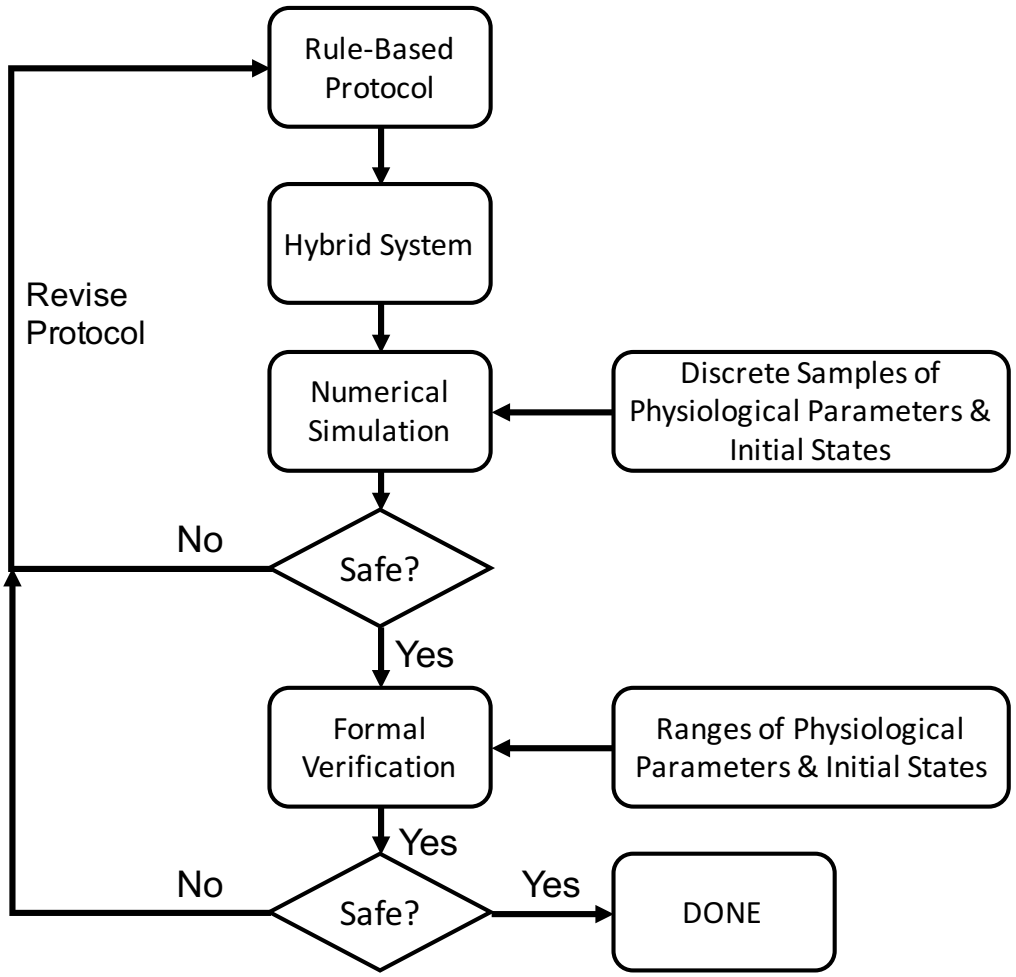


Figure 3.2: An iterative model-based analysis framework to evaluate, improve, and verify protocol-driven behaviors.

approach is that a "fail safe" mode exists, i.e., the system can always fall back to a pre-determined safe action such as stopping infusion. This assumption does not universally apply to all physiological systems. Verifying physiological systems that contain nonlinear dynamics, uncertain parameters and initiate states with no default safe mode remains a challenging problem.

We propose an iterative model-based analysis approach to evaluate, improve, and verify the safety of protocol-driven behaviors using physiological models that may contain nonlinearities, unidentifiable parameters, and unknown initial states. Figure 3.2 presents an overview of the framework. Our approach integrates numerical

simulation and formal verification by harnessing their relative strengths in analyzing physiological models. In our framework, a rule-based protocol is first evaluated in simulation on a set of virtual subjects. Simulation allows fast prototyping, and it is particularly useful at efficiently ruling out improper protocol designs during the early development stages: A protocol that is unsafe even for sample virtual subjects during finite-time simulation is unlike to be safe to be tested on humans. If a protocol fails simulation test, the simulated trajectories may provide insights into how the protocol design can be improved. After the simulation test, successful candidate protocols are further evaluated in formal verification. The key point is that verification provides safety guarantees with respect to model uncertainties: If a protocol is safe on certain regions of parameters and initial states, then it is safe for any patient that maps into those regions, even though the exact individual parameters and initial states may not be identifiable. The cost is that formal verification can be much more computationally expensive than simulation.

3.3 An Inpatient Glucose Control Case Study

In this section, we apply the proposed model-based framework to a clinical case study. This section is organized as follows: Section 3.3.1 introduces the application background and explains why model-based analysis is needed; Section 3.3.2 summarizes the contributions of this case study; Section 3.3.3 describes modeling a current Intra-Operative Glucose Control (IOGC) protocol and validating the model using a clinical dataset; in Section 3.3.4, we evaluate the IOGC model by simulation, identify its weaknesses, propose an improved protocol design, and validate that the new protocol overcomes the weaknesses of the current one while preserving its strengths in simulation-based virtual clinical trials; Section 3.3.5 formally verifies the safety of the new IOGC protocol design using an FDA-accepted physiological model; Section 3.3.6 proposes a novel technique for predictive safety monitoring in the presence

of intra-subject run-time physiological variances.

3.3.1 Motivation

For the more than 29 million Americans who have diabetes, the risk of death is nearly twice as high when compared to age-matched non-diabetic individuals [104, 119, 101, 281]. Those suffering from this disease, especially Type 1 diabetics, depend on insulin self-injections to manage their blood glucose level. As such, glucose regulation is a safety-critical control task: Too much insulin causes life-threatening hypoglycemia (low glucose levels), and too little insulin causes hyperglycemia (high glucose levels), a condition that has severe adverse outcomes such as blindness and nerve damage.

Hyperglycemia, unless iatrogenic, typically represents a secondary manifestation (i.e., epiphenomenon) of myriad physiologic, pharmacologic and/or metabolic derangements. While glucose is essential to life, in excess it is associated with increased cardiovascular morbidity and mortality in both diabetics and non-diabetics [153, 190]. It is therefore not surprising that the prevalence of diabetes mellitus and its associated complications among hospitalized patients are increasing [61, 264, 176]. While outpatient management of hyperglycemia has historically been the primary focus in this population, and has unquestionably reduced diabetic morbidity and mortality, mounting evidence suggests that inpatient glycemic control may impart a similar benefit [67, 102, 253, 18, 44, 193, 209, 207, 259].

More recent investigations have begun focusing efforts towards reducing hyperglycemia specifically among critically ill and perioperative patients. The primary genesis of this approach stems from a 2001 randomized, controlled study that reported significant decreases in ICU and hospital mortality when blood glucose levels (BGLs) were maintained between 80 – 100 mg/dL compared to a less aggressive level of 180 – 200 mg/dL [261]. Subsequent enthusiasm for aggressive glucose management, however, has tempered as multiple groups attempting to replicate those results were unable to show comparable reductions in morbidity and mortality and have

consistently observed high rates of hypoglycemia [229, 92, 43, 13, 78, 30]. Whether the lack of mortality benefit in these studies can be directly attributed to the high rates of hypoglycemia is uncertain. Many groups that have effectively maintained euglycemia while avoiding hypoglycemia have demonstrated improved outcomes in those treated more aggressively with insulin [161, 189, 173, 256]. Furthermore, wide and frequent oscillations in plasma glucose (so called, glycemic variability) appear to be as, if not more, important as absolute glucose values in critically ill patients and may compound any deleterious effects of hyperglycemia [87, 8, 83, 154].

While the appropriate target of plasma glucose in critically ill or perioperative patients remains elusive, there is general consensus with regards to three points: 1) Profound and sustained hyperglycemia in critically ill patients is likely harmful; 2) Isolated or sustained hypoglycemia in critically ill patients is likely harmful; 3) Wide and frequent variations in serum glucose values are likely harmful. As these three goals, nebulous as they may be, appear to be recurrent and unifying themes, attention must be directed towards methods, protocols and/or devices that can aid in achieving all three. Until accurate and reliable continuous glucose monitors are available for critically ill patients, we remain limited in our ability to measure, respond, and predict future glucose values. Protocols that take into account prior glucose readings and rates of change have been more successful but are far from foolproof [40, 39, 216, 198, 249, 12, 116].

Designing a protocol to reliably achieve glucose control, particularly when faced with frequent changes in physiologic parameters such as insulin sensitivity (as is seen perioperatively) is extremely challenging [162, 52, 93]. Current protocols are mostly derived from experience and intuition and are developed by local consensus. Unlike many engineering systems (e.g., electronic circuits and automobiles), where first-principle plant models can be derived from classical physics, it is extremely difficult to identify a mathematical model that would accurately predict the glucose-insulin dynamics of an individual with only limited measurable clinical data [114].

Furthermore, it is neither feasible nor ethical to test all potential insulin protocols in human patients.

While attempts to model glucose metabolism using computer simulation were first proposed in the 1960s, newer simulations that incorporate data-driven plant modeling of glucose metabolism now exist and are able to more accurately mimic glucose regulation in diabetic patients [118, 53, 126]. Repeated evaluations of numerous insulin infusion protocols, all slightly different, on live patients are neither realistic nor feasible. In-silico evaluation and simulation is a well accepted and validated means to examine a large number of iterative changes and is only recently being used to evaluate insulin infusion protocols [170].

3.3.2 Contributions

Using the proposed iterative analysis approach, we identify the weaknesses of a current intraoperative glycemic control (IOGC) protocol and design a new protocol that overcomes the weaknesses in simulation-based virtual clinical trials. We then formally verify the safety of the new protocol using a recently developed hybrid system model checker and demonstrate that the new protocol is safe over continuous spaces of parameters and initial states on an FDA-accepted advanced glucose metabolism model, the T1DMS model [73]. The safety requirements for insulin protocols are from established clinical consensus: Hypoglycemia (usually defined as glucose level less than 70 mg/dL [15]) can be life-threatening and severe hyperglycemia (usually defined as glucose level in the high range of more than 200 mg/dL [260]) have long-term complications. To the best of our knowledge, this is the first work on formally verifying insulin protocols using the T1DMS model that is initiated to continuous uncertain ranges of parameters and initial states. The verification results provide stronger safety evidence than other existing work along this line of research that rely on simulation over finite samples of virtual subjects [277, 182]. To further cope with the practical challenge that a patient's physiological parameters may experience

transient fluctuations during surgery, we propose a novel run-time predictive safety monitoring technique that leverages a maximal model coupled with online training of a computational virtual subject (CVS) set. To the best of our knowledge, the idea of using CVS for data-driven adaptive safety monitoring has not been explored before.

3.3.3 Protocol Modeling and Validation

By collaborating with clinicians at the Hospital of the University of Pennsylvania, we accessed and investigated a paper-based Insulin Infusion Protocol (referred to as IIP in the rest of the writing) that is currently used for cardiac bypass surgery patients. The IIP (as shown in Figure 3.3) consists of two parts: 1) a table that categorizes the BGL into a finite number of intervals and, based upon the current interval, sets a fixed intravenous bolus and infusion rate; 2) a set of infusion rate adjustment rules that take into account the relative change in BGL with respect to the previous value (Figure 3.3). The target BGL defined by the IIP is 70 – 130 mg/dL.

The IIP conforms to the rule-based protocol $\langle w, \mathbf{y}, \mathbf{u}, L\rangle$ in Definition 1. The sampling period w is 30 minutes. At the k-th sample point, it has two monitoring variables $\mathbf{y} = [y(k), y(k-1)]$: Current BG, $y(k)$, and the previous BG, $y(k-1)$. There are two control variables $\mathbf{u} = [u_b(k), u_c(k)]$: Insulin bolus $u_b(k)$ and insulin infusion rate $u_c(k)$. We apply the hybrid system model in Definition 3 to representing the clinicians' behaviors as guided by the IIP.

Definition 4 *Given the IIP* $\langle w, \mathbf{y}, \mathbf{u}, L\rangle$ *(Figure 3.3) and the glucose physiological process represented as a function* $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u})$*, the hybrid system model of the closed-loop system is* $\mathcal{H} = \langle \mathcal{X}, \mathcal{Q}, \mathcal{X}_{init}, \mathcal{X}_{inv}, \mathcal{F}, T\rangle$*:*

- $\mathcal{X} = \{t, \mathbf{y}, \mathbf{u}\}$.
- $\mathcal{Q} = \{Q_0\}$.

Target Glucose: 70 - 130 mg/dL

**** Glucose must be checked every 30 minutes ****

*** INSULIN Bolus / Infusion Protocol ***

Initiation of Protocol

Initiate protocol if *any one* of the following criteria exist:

- Previous diagnosis of diabetes mellitus
- Any blood glucose (BG) > 120 mg/dL
- Any patient arriving to operating room on I.V. Insulin
- Anticipated administration of steroids
- Planned circulatory arrest

Blood Glucose (mg/dL)	Insulin Bolus (U)* **(No bolus pre-CPB)**	Insulin Infusion (U/h)
< 100	0	0
100 – 110	0	2
111 - 130	0	4
131 – 150	2	4
151 – 170	4	6
171 – 190	4	8
191 – 210	6	8
211 – 230	8	10
231 – 250	10	10
251 – 300	12	14
> 300	15	15

INSULIN *TITRATION* PROTOCOL (start after INITIATING insulin infusion)

*** If BG unchanged- repeat action on Infusion Protocol ***

Blood Glucose (mg/dL)	Action
< 60	**25 mL of D_{50} I.V. AND STOP ALL INSULIN**
60 – 99	• If BG ↓ by 30 mg/dL or less from last BG, stop infusion • If BG ↓ by greater than 30 mg/dL from last BG, 25 mL of D_{50} I.V. • If BG ↑ from last BG, NO infusion and NO bolus
100 – 150	BG Less than Prior • If BG ↓ by 30 mg/dL or less from last BG, ↓ infusion by 2 U/h and NO bolus • If BG ↓ by greater than 30 mg/dL from last BG, ↓ infusion by 4 U/h and NO bolus BG Greater than Prior • If BG ↑ by 10 mg/dL or less from last BG, continue infusion with ½ bolus • If BG ↑ by greater than 10 mg/dL from last BG, continue per infusion protocol
151 – 170	BG Less than Prior • If BG ↓ by 30 mg/dL or less from last BG, continue per infusion protocol, NO bolus • If BG ↓ by greater than 30 mg/dL, start ½ recommended infusion, NO bolus BG Greater than Prior • If BG ↑ by 10 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↑ by greater than 10 mg/dL from last BG, continue per infusion protocol
171 – 200	BG Less than Prior • If BG ↓ by 30 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↓ by greater than 30 mg/dL, continue per infusion protocol, NO bolus BG Greater than Prior • If BG ↑ by 10 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↑ by greater than 10 mg/dL from last BG, continue per infusion protocol
201 – 250	BG Less than Prior • If BG ↓ by 30 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↓ by greater than 30 mg/dL, continue per infusion protocol, NO bolus BG Greater than Prior • If BG ↑ by 10 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↑ by greater than 10 mg/dL from last BG, continue per infusion protocol
251 – 300	BG Less than Prior • If BG ↓ by 30 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↓ by greater than 30 mg/dL, continue per infusion protocol, NO bolus BG Greater than Prior • If BG ↑ by 10 mg/dL or less from last BG, continue per infusion protocol with ½ bolus • If BG ↑ by greater than 10 mg/dL from last BG, continue per infusion protocol
> 300	Continue per infusion protocol

Figure 3.3: The insulin infusion protocol. Abbreviations: BG, Blood glucose; CPB, Cardiopulmonary bypass; D50, 50 percent Dextrose (50 gram/100 mL).

- $\mathcal{X}_{init}(Q_0) = True.$

- $\mathcal{F}(Q_0) = \{\dot{t} = 1, \dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}), \dot{\mathbf{u}} = 0\}$.
- $\mathcal{X}_{inv}(Q_0) = \langle t \leq w + \delta \rangle$.
- $T = \bigcup_{i=1}^{n} \{T_i\}$, *where* T_i *denotes the self-transition that corresponds to the i-th rule:* $T_i = (t \geq w - \delta \wedge G_i(\mathbf{y}) = True \xrightarrow{Q_0 \to Q_0} \mathbf{u} = \mathbf{u}^i \wedge t = 0)$. *Each rule in the IIP is encoded as a self-transition, e.g., the first rule in the "BG in 60-90 mg/dL" box is encoded as* $T_i = (t \geq 30 - \delta \wedge y(k) \in [60, 99] \wedge y(k) - y(k-1) \leq 30 \xrightarrow{Q_0 \to Q_0} u_b(k) = 0 \wedge u_c(k) = 0 \wedge t = 0)$.

To validate the behavior model, we simulate it using an FDA-accepted high-fidelity physiological model and compare the simulated glucose measurements with a clinical glucose dataset that is collected from patients who were on IIP. After obtaining the acknowledgment from the University of Pennsylvania Institutional Review Board (IRB), blood glucose measurements were retrospectively evaluated on 57 type 1 diabetic patients controlled with the IIP during the period of cardiopulmonary bypass.

We use the T1DM Simulator [75], which is developed in Matlab/Simulink®. The patient glucose model that it utilizes is based on a high-dimensional, non-linear differential equation model [73, 151]. The T1DM Simulator (academic version) comes with 10 pre-identified Type 1 Diabetic "virtual" adult subjects. Each virtual subject is a realization of the patient-specific parameters that are used by the simulation model (e.g., body weight and insulin/glucose transportation rates between different body compartments). Many of these parameters cannot be directly identified from the clinical data that hospitals currently have (e.g., total insulin/glucose distribution volumes). The virtual population in the software was identified based on laboratory data collected from a group of individuals who participated in a triple-tracer meal experiment [151]: The meals are marked with isotope tracers so that the glucose/insulin fluxes in the body can be measured. In 2008, the T1DM Simulator received FDA approval for computer simulations that could be substituted for animal trials

in pre-clinical testing and has become an accepted method of evaluation for studies in patients with type 1 diabetes mellitus [225]. The simulator is a Simulink® model file within MATLAB® that includes the patient model, glucose sensor and insulin pump models, and an interface for user-defined controllers [221]. We implement the hybrid system behavior model in Simulink using the Stateflow® toolbox.

Using the T1DMS model, we simulate the protocol-based behavior model on 10 virtual patients. The experiments are repeated with different initial BGL values to more thoroughly investigate the protocol's performance. Blood glucose measurements are taken every 30 minutes (as defined in the protocol). For each real patient's measured BGL trajectory, we run closed-loop simulation on the 10 virtual subjects starting from the same initial BGL and pick the virtual subject whose simulated BGL trajectory best matches the measured BGLs, i.e., had the lowest maximum deviation. Simulated BGL data is then compared with the actual BGL data measured.

We validate the behavior model by comparing the key metrics of the simulated BGLs and the real patients' BGLs. In addition to comparing the mean and standard deviation of the per-subject BGL values, we calculate the normalized glucose lability index (NGLI, $[\text{mg/dL}]^2/\text{hour}^2$), akin to the weekly glucose lability index (GLI) [237]. The GLI $\sum_{i=1}^{N(\text{of 1 week})} \frac{(Glu_{i+1}-Glu_i)^2}{T_{i+1}-T_i}$ is a measure of the weekly sum of the rate of change of BGL, where Glu_i is the i-th glucose reading (mg/dL) taken at time T_i. Because the length of surgery is different for each patient, the GLI must be normalized to the total length of the measurement time, $T_N - T_1$, where N is the total number of glucose values obtained for the patient (dependent on the length of surgery), T_N is the final time of the measurement period and T_1 is the initial time. We define this normalized metric as the NGLI. The NGLI was thus calculated as follows:

$$NGLI = \frac{\sum_{i=1}^{N} \frac{(Glu_{i+1}-Glu_i)^2}{T_{i+1}-T_i}}{T_N - T_1}$$

A higher NGLI implies that the BGL trajectory exhibits more variability.

Table 3.1 presents the summative data for validation of the virtual subjects.

Table 3.1: Comparison of the effect of IIP on BGL in 10 virtual patients in-silico with those of IIP on BGL in real patients. Abbreviations: IIP, Insulin infusion protocol; STD, standard deviation; BGL, blood glucose level; NGLI, normalized glucose lability index; NS, not significant. Mean values were compared via two-tailed unpaired t-test.

	Real Patients (n = 57)	Virtual Patients (n = 10)	p
Average per-subject Mean STD of BGL (mg/dL)	130 ± 16.0	114 ± 15.3	0.0047
Average per-subject NGLI ($[mg/dL]^2/hr^2$)	1775	1782	NS

This represents the effect of the IIP on the virtual patients compared with the retrospectively observed data in the actual 57 patients. Of note, the T1DM Simulator does not and cannot account for all parameters, both in-vivo (i.e., time varying insulin sensitivity) and ex-vivo (i.e., the effect of cardiopulmonary bypass). However, as seen in Table 3.1, the standard deviations of BGL values and NGLIs of the two groups (virtual and real) are similar. Thus, the virtual population is able to closely reproduce the BGL variability observed in the real data.

While the standard deviations and NGLIs are quite similar, the per-subject means in the two populations are significantly different. We believe this is due to several reasons. First, the initial physiological states of the two populations are possibly mismatched, due to the unobservable physiological states challenge described in Section 3.2.2. Each virtual patient needs an initial configuration that includes all physiological states at simulation time zero, which defines the initial condition for the differential equations that describe the patient model. Most of these physiological states are not directly measurable (i.e., the total mass of glucose and insulin in different compartments). In our experiments, these physiological variables were set to be started in a stable state. Mathematically this means that the initial conditions of the differential equations are solved by setting all derivatives to zero. However, such exquisite homeostasis is unlikely to be present in a real patient at the start of surgery. An additional reason for the observed mean inter-subject variability is that

we have a limited quantity of real population data from which we could draw conclusions. Intraoperatively, the BGL was measured at a relatively low frequency (every 30 minutes), and therefore, for each individual patient, there are usually fewer than 10 BGL readings over the entire surgery. If the initial states of the two populations are mismatched, the virtual population may not be able to converge to the state of the real patients within such limited time and with such few measurements. Furthermore, while the IIP dictates measuring the BGL every 30 minutes, the reality is that sampling unlikely occurred exactly every 30 minutes in our actual patient population. Finally, we use a relatively small group size for our virtual patient population. The "best match" for each patient could only be chosen from 10 virtual subjects and may not include a good match for each real patient. This would explain why the population standard deviation and NGLI match better than individual data.

3.3.4 Protocol Evaluation and Enhancement by Simulation

When evaluating the behavior model (see Definition 4) on the in-silico population, the simulation time length for each in silico experiment is 24 hours. While the typical cardiac surgical procedure usually takes only 3 – 4 hours, running the simulation for longer periods revealed the "stable" control pattern. A pattern typical of all controllers is an initial variability around the set target before stability. It is essential to evaluate for an extended period to ensure prolonged stability. Each BGL trajectory was then divided into two epochs: 0 – 5 hours ("initial" phase) and 12 – 24 hours ("oscillating" phase). By adjusting the initial BGL in all 10 virtual patients, we evaluate the efficacy of the IIP.

Key metrics of the simulated BGL trajectories on the 10 virtual subjects in response to the IIP are shown in Table 3.2. For each initial BGL, a simulation run generates 10 BGL trajectories (from 10 virtual subjects) and metrics are reported for the initial (0 – 5 hours) phase and the oscillating (12 – 24 hours) phase. This data illustrates that during the oscillating phase, the IIP is able to keep most BGLs

Table 3.2: Key metrics of simulated BGL controlled by the IIP in virtual patients (n=10)

	Simulated BGL (0 – 5h)						
Init BGL (mg/dL)	Mean BGL (mg/dL)[1]	STD of BGL (mg/dL)[1]	NGLI ($[mg/dL]^2/h^2$)[1]	70 – 130[2]	> 130[2]	< 70[2]	Min BGL (mg/dL)[1]
70	86[80,92]	15[9,18]	411[55,1533]	99%[90%,100%]	0%[0%,0%]	1%[0%,10%]	70[66,70]
80	93[89,98]	12[9,16]	286[48,1253]	99%[90%,100%]	0%[0%,0%]	1%[0%,10%]	78[63,80]
90	99[97,102]	8[5,19]	216[16,915]	99%[90%,100%]	1%[0%,10%]	0%[0%,0%]	88[78,90]
100	98[88,101]	7[2,15]	318[9,1400]	99%[90%,100%]	0%[0%,0%]	1%[0%,10%]	89[68,98]
110	104[97,107]	6[2,16]	388[10,1781]	99%[90%,100%]	0%[0%,0%]	1%[0%,10%]	95[69,105]
120	102[91,111]	12[5,22]	789[24,3044]	95%[80%,100%]	0%[0%,0%]	5%[0%,20%]	85[60,106]
130	107[96,119]	14[6,25]	928[37,3278]	97%[80%,100%]	0%[0%,0%]	3%[0%,20%]	90[58,112]
140	104[91,117]	17[11,28]	1314[112,4226]	84%[60%,90%]	11%[10%,20%]	5%[0%,30%]	85[53,106]
150	107[93,125]	20[12,31]	1719[129,5728]	81%[60%,90%]	14%[10%,20%]	5%[0%,30%]	87[55,113]
160	101[85,113]	26[20,36]	2734[415,7501]	80%[50%,90%]	12%[10%,20%]	8%[0%,40%]	74[41,98]
170	103[87,115]	29[22,38]	2992[605,8214]	79%[50%,90%]	14%[10%,20%]	7%[0%,40%]	75[42,99]
180	103[88,112]	34[27,41]	3633[778,8644]	73%[50%,80%]	17%[10%,30%]	10%[0%,40%]	70[38,90]
190	106[89,119]	36[30,43]	3984[860,9464]	74%[50%,80%]	17%[10%,30%]	9%[0%,40%]	72[39,95]
200	105[89,130]	39[31,48]	5310[966,11978]	71%[50%,80%]	17%[10%,30%]	12%[0%,40%]	69[32,108]
210	106[91,118]	43[36,50]	5829[1435,13084]	71%[50%,80%]	17%[10%,30%]	12%[0%,40%]	67[33,94]
220	101[89,115]	49[40,56]	7301[2065,15537]	65%[40%,80%]	17%[10%,30%]	18%[0%,40%]	57[26,85]
230	105[92,117]	51[43,59]	8207[2243,16927]	69%[50%,80%]	17%[10%,30%]	14%[0%,40%]	60[27,88]
240	102[89,117]	55[46,62]	9286[2788,18936]	63%[40%,80%]	17%[10%,30%]	20%[0%,50%]	56[24,84]
250	105[91,121]	58[49,65]	10204[3010,21115]	63%[40%,80%]	17%[10%,30%]	20%[0%,50%]	57[24,87]
	Simulated BGL (12 – 24h)						
70	104[91,108]	10[4,29]	882[19,5088]	94%[58%,100%]	2%[0%,17%]	4%[0%,25%]	88[60,101]
80	104[91,108]	10[2,27]	878[9,5083]	92%[38%,100%]	2%[0%,21%]	6%[0%,42%]	89[64,104]
90	104[92,108]	10[1,30]	877[4,5045]	94%[54%,100%]	2%[0%,21%]	4%[0%,25%]	87[60,105]
100	104[91,109]	10[1,31]	927[4,5782]	92%[42%,100%]	2%[0%,21%]	6%[0%,38%]	89[59,106]
110	104[91,111]	10[3,31]	951[12,5707]	93%[42%,100%]	2%[0%,21%]	5%[0%,38%]	87[59,100]
120	104[89,109]	10[1,29]	889[8,5030]	93%[54%,100%]	2%[0%,21%]	5%[0%,25%]	88[59,105]
130	104[89,109]	10[2,30]	879[3,4997]	93%[46%,100%]	2%[0%,21%]	5%[0%,33%]	89[59,106]
140	104[94,109]	10[2,29]	903[10,5411]	94%[50%,100%]	2%[0%,21%]	4%[0%,29%]	89[59,101]
150	104[91,111]	10[2,31]	962[4,5690]	92%[42%,100%]	2%[0%,21%]	6%[0%,38%]	89[59,103]
160	104[91,110]	10[1,29]	921[4,5441]	93%[50%,100%]	2%[0%,21%]	5%[0%,29%]	89[59,106]
170	104[91,109]	10[2,29]	901[9,5410]	93%[50%,100%]	2%[0%,21%]	5%[0%,29%]	89[59,105]
180	106[97,109]	10[2,29]	984[8,5315]	95%[54%,100%]	2%[0%,21%]	3%[0%,25%]	89[59,104]
190	105[96,110]	10[2,32]	1049[15,5867]	94%[46%,100%]	2%[0%,21%]	4%[0%,33%]	89[59,105]
200	104[90,108]	10[2,32]	972[12,5804]	93%[46%,100%]	2%[0%,21%]	5%[0%,33%]	88[58,103]
210	104[90,108]	10[2,32]	954[8,5767]	93%[46%,100%]	2%[0%,21%]	5%[0%,33%]	89[58,105]
220	105[95,109]	9[2,28]	868[10,4954]	95%[63%,100%]	1%[0%,13%]	3%[0%,25%]	89[58,106]
230	105[95,108]	10[1,28]	874[2,4967]	95%[63%,100%]	1%[0%,13%]	3%[0%,25%]	89[58,107]
240	105[91,109]	9[2,27]	869[13,5005]	94%[63%,100%]	2%[0%,17%]	4%[0%,21%]	91[58,105]
250	105[92,109]	9[1,27]	849[6,5005]	95%[63%,100%]	2%[0%,17%]	4%[0%,21%]	90[58,105]

All data are presented on a per-trajectory basis. Each row represents a starting BGL value and data from 10 trajectories (each virtual patient). [1]Data represent Mean[Minimum, Maximum]. Target range for IIP was 70 – 130 mg/dL. [2]Values represent Mean % [minimum %, maximum %] BGL within stated range for each trajectory. Abbreviations: BGL, blood glucose level; STD, standard deviation; NGLI, normalized glucose liability index; Min, minimum BGL. Example: when initial BGL is 70 mg/dL and simulation is run using IIP on the 10 virtual patients, the mean value for the mean BGL of each trajectory (virtual patient) was 86 mg/dL. The virtual patient with the lowest mean had a mean BGL of 80 mg/dL and the trajectory with the highest mean had a mean of 92 mg/dL. Similar interpretations can be made for the STD BGL, NGLI and Min. Within this same trajectory, the mean % of BGLs that were within target was 99%. At the lowest end, 90% of the BGLs were within target and at the highest end 100% of the BGLs were within target. Similar interpretations can be made for the > 130 and < 70 mg/dL ranges.

within the target range (70 – 130 mg/dL) for most subjects. Interestingly, given that the minimum of in-target percentage during this phase is less than 60% in most runs, it is also apparent that there exist subjects whose BGL's do not track the target well. On the contrary, in the initial phase, the quality of target tracking with IIP is very dependent on the initial BGL. However, when the initial BGL is high, the number of in-target BGLs decrease and the number of below-target (< 70

BGL (n) (mg/dL)	Action
≤ 60	Give 25 mL D_{50}
60 - 300	Rate (U/hr) = max(0, K_p[BGL(n) – Target] + K_D[BGL(n) – BGL(n-1)] + R_B) **IF** [BGL(n) – BGL(n-1)] < -30 **AND** BGL(n) < 100 **THEN** Rate= 0 and give [(BGL(n-1) – BGL(n))x 0.2] mL D_{50}
> 300	Rate (U/hr) = 15 and give 15 U Insulin bolus

Figure 3.4: A Proportional-Derivative Protocol for controlling blood glucose intraoperatively. Abbreviations: BGL Blood glucose level; U Units; D50 50 percent Dextrose (50 g/100 mL); BGL(n) current blood glucose reading; BGL(n-1) previous blood glucose reading; K_P Proportional gain (U/hr per mg/dL; after tuning= 0.05); K_D Derivative gain (U/hr per mg/dL; after tuning=0.06); Target Blood glucose target (set to 100 mg/dL); R_B Basal insulin rate (U/hr; after tuning= 1.0).

mg/dL) BGLs increase significantly. This is typical of many currently used insulin infusion protocols and results from the "overshooting" phenomenon. That is, most protocols start with a high dose of insulin bolus and infusion when the initial BGL is high, resulting in the subsequent BGL going below the target range. The resultant hypoglycemia is caused by two primary factors: 1) the overshooting during the initial phase, and 2) extreme oscillation in the oscillating phase. In the initial phase, when the initial BGL is greater than 130 mg/dL, overshooting causes a significant percent (as high as 50%) of low BGL readings. In the oscillating phase, when considering out-of-target BGL readings, there are more low BGL readings (as high as 42%) than high BGL readings. Thus, even when the control pattern stabilizes, the risk of hypoglycemia remains significant with the IIP for some subjects.

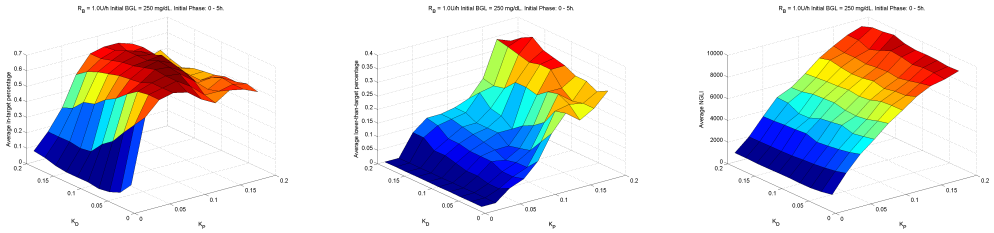
After evaluating the weaknesses and strengths of the IIP in silico, we design a new Proportional-Derivative Protocol (PDP), as described in Figure 3.4. The PDP is designed so that the sampling period would be the same as the IIP (30 minutes) and only the calculation method of insulin and dextrose dose would change. Similar to the IIP, the PDP only makes use of the current and previous BGL readings so that caregivers do not have to collect any additional information to implement the

PDP in the real clinical environment. For safety purposes, the PDP retains the same fixed actions of the IIP if and when the BGL reaches an extreme (e.g., BGL $<$ 60 mg/dL or BGL $>$ 300 mg/dL). The critical part of PDP is that when the BGL is in the control zone (60 – 300 mg/dL), the intravenous insulin infusion rate is calculated and changed in response to a proportional-derivative law, where K_P, K_D, Target, and R_B are the proportional gain (U/hr per mg/dL), derivative gain (U/hr per mg/dL), target value (mg/dL), and basal insulin rate (U/hr), respectively. Since one cannot administer "negative" insulin, when the rate calculated by the PD law is less than zero, we set the rate to zero. Additionally, if BGL drops too fast (defined as $BGL(n) - BGL(n-1) < -30$) and the current BGL(n) is already below 100 mg/dL, we stop the insulin infusion and give intravenous dextrose to counteract impending hypoglycemia. The amount of dextrose (D_{50}) administered is proportional to the magnitude of BGL decrease using the derivative law.

The control gains must be tuned such that performance, however it is defined, can be optimized. For this study, we pick three performance metrics to monitor and optimize: 1) the percentage of BGL values in the target range (70 – 130 mg/dL), 2) the percentage of BGL values lower than the target range ($<$ 70 mg/dL), and 3) NGLI. These represent the quality of target tracking, hypoglycemia risk control, and variability minimization, respectively. Additionally, from a clinical standpoint, when tuning the protocol, we consider hypoglycemia to be the primary safety concern. Thus, if a trade-off has to be made between very low BGL and higher than target BGL, we favor the latter.

There are classic control theory methods that allow one to analytically calculate the optimal control gains based on the mathematical model of the control plant. However, it is very difficult to apply the analytical methods to the patient simulator that we are using, because the model is highly nonlinear (most classic control approaches assume linear plant models), and many model variables cannot be directly measured in real time on general patients (the tracer experiment used to identify

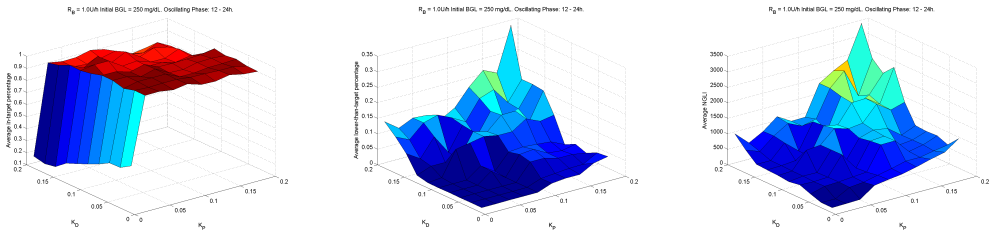
the virtual subjects is clearly too resource demanding to be applied to all ICU patients). Therefore, we tune the protocol parameters based on numerical simulations. To identify the optimal setting of the PDP parameters, we systematically vary K_P, K_D, and R_B on a wide range of values and examine how the performance metrics are impacted by different settings. The optimal setting of K_P, K_D, and R_B is the one such that the three performance metrics (in-target percentage, lower-than-target percentage, and NGLI) are optimized.



(a) Initial (0 – 5 hr) phase. R_B = 1.0 U/hr and initial BGL = 250 mg/dL. Metric= in-target (70 – 130 mg/dL) percentage.

(b) Initial (0 – 5 hr) phase. R_B = 1.0 U/hr and initial BGL = 250 mg/dL. Metric= lower-than-target (< 70 mg/dL) percentage.

(c) Initial (0 – 5 hr) phase. R_B = 1.0 U/hr and initial BGL = 250 mg/dL. Metric= NGLI.



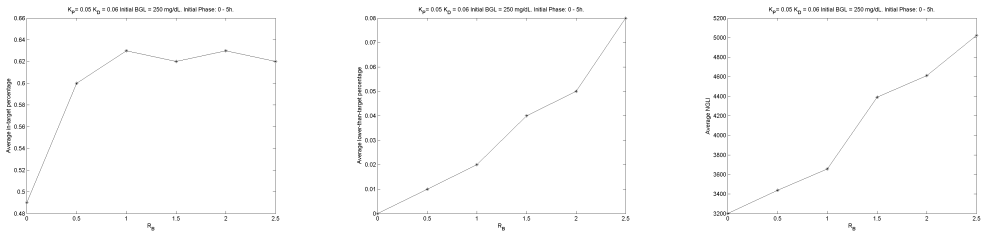
(d) Oscillating (12 - 24 hr) phase. R_B = 1.0 U/hr and initial BGL = 250 mg/dL. Metric= in-target (70 – 130 mg/dL) percentage.

(e) Oscillating (12 – 24 hr) phase. R_B = 1.0 U/hr and initial BGL = 250 mg/dL. Metric= lower-than-target (< 70 mg/dL) percentage.

(f) Oscillating (12 – 24 hr) phase. R_B = 1.0 U/hr and initial BGL = 250 mg/dL. Metric= NGLI.

Figure 3.5: Impact of K_P and K_D. Abbreviations: K_P Proportional gain (U/hr per mg/dL); K_D Derivative gain (U/hr per mg/dL); R_B Basal insulin rate (U/hr).

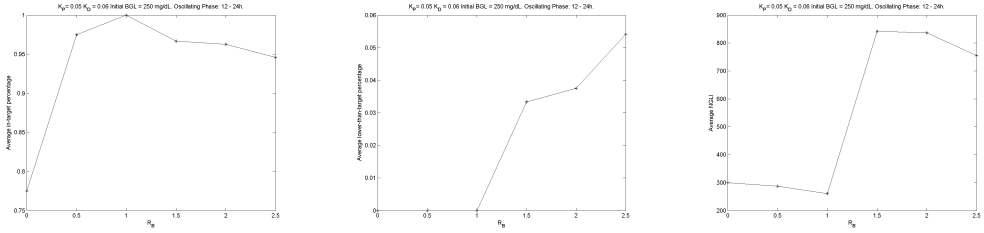
We choose the target for the controller to be fixed at 100 mg/dL, which represents a value in the middle of the IIP target range (70 – 130 mg/dL) and is also the insulin action start point in the IIP. The impact of incremental changes of K_P and K_D on our three primary metrics, both for the initial phase (0 – 5 hours) and oscillating



(a) Initial (0 – 5 hr) phase. Initial BGL = 250 mg/dL. Metric= In-target (70 – 130 mg/dL) percentage.

(b) Initial (0 – 5 hr) phase. Initial BGL = 250 mg/dL. Metric= Lower-than-target (< 70 mg/dL) percentage.

(c) Initial (0 – 5 hr) phase. Initial BGL = 250 mg/dL. Metric= NGLI.



(d) Oscillating (12 – 24 hr) phase. Initial BGL = 250 mg/dL. Metric= In-target (70 – 130 mg/dL) percentage.

(e) Oscillating (12 – 24 hr) phase. Initial BGL = 250 mg/dL. Metric= Lower-than-target (< 70 mg/dL) percentage.

(f) Oscillating (12 – 24 hr) phase. Initial BGL = 250 mg/dL. Metric= NGLI.

Figure 3.6: Impact of R_B on performance metrics after optimal tuning of PD parameters (K_P = 0.05, K_D = 0.06). Abbreviations: K_P Proportional gain (U/hr per mg/dL); K_D Derivative gain (U/hr per mg/dL); R_B Basal insulin rate (U/hr).

phase (12 – 24 hours) are shown in Figure 3.5. Simulation results show that R_B, in its variance range, does not significantly change the shapes of the K_P-K_D performance surfaces shown in Figure 3.5, in which $R_B = 1.0$ U/hr. Therefore, we first identify the optimal K_P-K_D setting by integrating the three performance metrics. Key findings in this analysis include that hypoglycemia (< 70 mg/dL) and NGLI are minimized when K_P and K_D are relatively small. The in-target (70 – 130 mg/dL) percentage is maximized when K_P and K_D are in the lower middle range (see Figure 3.5 and Table 3.3). Integrating the data analysis with the performance metrics, we identify the optimal K_P-K_D setting, $K_P = 0.05$ U/hr per mg/dL and $K_D = 0.06$ U/hr per mg/dL, in the region where the peak areas of in-target percentages (Figures 3.5 (a) and (d)) overlap the low areas of lower-than-target range and NGLI (Figures 3.5 (b),

Table 3.3: Observations noted when tuning the PD controller.

In-target (70 – 130 mg/dL) BGLs	
K_P	Initial phase: As K_P increases, metric first increases then decreases Oscillating phase: As K_P increases, metric decreases (except when K_D is large)
K_D	Initial phase: As K_D increases, the peak K_P increases Oscillating phase: As K_D increases, metric decreases
R_B	Initial phase: metric is maximized when $R_B \geq 1$ Oscillating phase: metric is maximized when $R_B \geq 1$
Less-than-target (< 70 mg/dL) BGLs	
K_P	Initial phase: As K_P increases, metric increases Oscillating phase: As K_P increases, metric increases when K_D is low
K_D	Initial phase: K_D is not the dominating factor Oscillating phase: As K_D increases, metric increases
R_B	Initial phase: As R_B increases, metric increases Oscillating phase: As R_B increases, metric increases
NGLI	
K_P	Initial phase: As K_P increases, metric increases Oscillating phase: As K_P increases, metric increases
K_D	Initial phase: K_D is not the dominating factor Oscillating phase: K_D is not the dominating factor
R_B	Initial phase: As R_B increases, metric increases Oscillating phase: As R_B increases, metric is minimized (when $R_B \leq 1$)

Abbreviations: PD, proportional derivative (controller); K_P, proportional gain (U/hr per mg/dL); K_D, derivative gain (U/hr per mg/dL); R_B, basal insulin rate (U/hr); BGL, blood glucose level; NGLI, normalized glucose liability index [$(\text{mg/dL})^2/\text{hr}^2$].

(c), (e), and (f)). The metrics are then further optimized by evaluating incremental increases in basal insulin rate (R_B) (Figure 3.6) according to the same performance metrics and the optimal PDP parameter setting is $K_P = 0.05$ U/hr per mg/dL, $K_D = 0.06$ U/hr per mg/dL, $R_B = 1.0$ U/hr.

Table 3.4 shows the performance metrics of the PDP on the virtual population. When comparing the PDP to the IIP (Table 3.2) there are some notable differences. With regard to target tracking, in the oscillating phase, the PDP is able to maintain close to 100% of the BGL readings within the target range. In the initial phase, when the BGL starts within the target range (70 – 130 mg/dL), the PDP is able to maintain almost 100% of BGL values in the target range. When the BGL starts > 130 mg/dL, the average in-target percentages are similar between the two algorithms. When comparing the risk of hypoglycemia between the IIP and the PDP, the PDP is noted to almost completely eliminate occurrences of BGL < 70 mg/dL in both

Table 3.4: Key metrics of simulated BGL controlled by the PD controller.

	Simulated BGL (0 – 5h)						
Init BGL (mg/dL)	Mean BGL (mg/dL)[1]	STD of BGL (mg/dL)[1]	NGLI ($[mg/dL]^2/h^2$)[1]	70 – 130[2]	> 130[2]	< 70[2]	Min BGL (mg/dL)[1]
70	85[79,91]	12[8,16]	76[36,163]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	70[70,70]
80	91[87,96]	9[6,12]	34[17,56]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	80[80,80]
90	95[92,99]	6[3,9]	33[8,114]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	89[85,90]
100	99[96,102]	5[1,9]	59[2,252]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	93[84,99]
110	103[97,106]	6[3,12]	121[5,546]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	95[82,103]
120	106[98,111]	8[6,14]	187[16,744]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	96[79,105]
130	109[99,116]	11[8,16]	272[39,905]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	97[77,106]
140	112[100,121]	14[12,17]	343[68,963]	83%[70%,90%]	17%[10%,30%]	0%[0%,0%]	98[81,108]
150	115[102,126]	18[15,20]	496[114,1371]	77%[60%,90%]	23%[10%,40%]	0%[0%,0%]	99[80,110]
160	118[103,131]	21[18,24]	662[166,1760]	76%[60%,90%]	24%[10%,40%]	0%[0%,0%]	99[78,111]
170	121[104,135]	25[22,27]	861[237,2211]	72%[50%,90%]	28%[10%,50%]	0%[0%,0%]	99[77,113]
180	123[105,139]	28[25,31]	1095[312,2796]	70%[50%,80%]	30%[20%,50%]	0%[0%,0%]	99[76,114]
190	126[105,144]	32[28,34]	1356[399,3383]	69%[40%,80%]	31%[20%,60%]	0%[0%,0%]	99[74,116]
200	128[106,148]	35[32,38]	1661[504,4129]	68%[40%,80%]	32%[20%,60%]	0%[0%,0%]	98[72,116]
210	131[106,152]	39[35,41]	1997[629,4874]	67%[40%,80%]	33%[20%,60%]	0%[0%,0%]	98[70,117]
220	133[108,156]	43[38,45]	2353[758,5576]	65%[40%,80%]	34%[20%,60%]	1%[0%,10%]	98[69,118]
230	135[108,160]	46[42,48]	2770[904,6468]	64%[40%,80%]	34%[20%,60%]	2%[0%,10%]	97[67,118]
240	138[109,164]	50[45,52]	3195[1062,7374]	63%[30%,80%]	35%[20%,70%]	2%[0%,10%]	97[65,119]
250	140[110,167]	53[49,56]	3656[1236,8272]	63%[30%,80%]	35%[20%,70%]	2%[0%,10%]	97[64,119]
	Simulated BGL (12 – 24h)						
70	110[98,116]	3[0,14]	255[0,1603]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[81,116]
80	110[99,116]	3[0,15]	230[0,1571]	100%[96%,100%]	0%[0%,4%]	0%[0%,0%]	106[81,116]
90	110[98,116]	3[0,15]	271[0,1744]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[81,116]
100	110[98,116]	3[0,15]	258[0,1731]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[80,116]
110	110[98,116]	3[0,15]	258[0,1632]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[82,116]
120	110[98,116]	3[0,15]	271[0,1779]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[82,116]
130	110[98,116]	3[0,14]	246[0,1526]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[82,116]
140	110[98,116]	3[0,15]	267[0,1745]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[82,116]
150	110[99,116]	3[0,14]	212[0,1484]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	107[86,116]
160	110[98,116]	3[0,15]	261[0,1689]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[83,116]
170	110[99,116]	3[0,15]	245[0,1562]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	105[78,116]
180	110[98,116]	3[0,14]	238[0,1450]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[81,116]
190	110[98,116]	3[0,15]	253[0,1600]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[81,116]
200	110[98,116]	3[0,15]	253[0,1613]	100%[96%,100%]	0%[0%,4%]	0%[0%,0%]	106[81,116]
210	110[98,116]	3[0,15]	267[0,1745]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[82,116]
220	110[98,116]	3[0,15]	257[0,1641]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	105[78,116]
230	110[98,116]	3[0,15]	249[0,1560]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	105[78,116]
240	110[98,116]	3[0,14]	239[0,1467]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	106[82,116]
250	110[98,116]	3[0,15]	261[0,1675]	100%[100%,100%]	0%[0%,0%]	0%[0%,0%]	105[80,115]

All data are presented on a per-trajectory basis. Each row represents a starting BGL value and data from 10 trajectories (each virtual patient). [1]Data represent Mean [Minimum, Maximum]. Target for PD controller was 100 mg/dL. [2]Values represent Mean % [minimum%, maximum%] BGL within stated range for each trajectory. Abbreviations: BGL, blood glucose level; STD, standard deviation; NGLI, normalized glucose liability index; Min, minimum BGL. Example: when initial BGL is 70 mg/dL and simulation is run using the PD controller on the 10 virtual patients, the mean value for the mean BGL of each trajectory (virtual patient) was 85 mg/dL. The virtual patient with the lowest mean had a mean BGL of 79 mg/dL and the trajectory with the highest mean had a mean of 91 mg/dL. Similar interpretations can be made for the STD BGL, NGLI and Min. Within this same trajectory, the mean % of BGLs that were within range of 70 – 130 mg/dL was 100%. There were no trajectories that fell out of this range. Similar interpretations can be made for the $>$ 130 and $<$ 70 mg/dL ranges.

phases, though this comes with a concomitantly greater frequency of BGL values $>$ 130 mg/dL in the initial phase compared with the IIP. Furthermore, the average minimum BGL achieved by the PDP is also significantly higher (but within the target-range), especially for the "extreme" subjects (the min minimum BGL, or lowest minimal BGL), thus further reducing the chance of hypoglycemia. Finally, with regards to BGL variability, the PDP achieves significantly lower NGLI than the

IIP (less than half the variability in both phases for most initial BGLs).

Simulation results clearly demonstrate that the PDP enhances IIP by overcoming its weaknesses while preserving its strengths. Similar to the IIP, when initial BGL values are within the target range, the PDP maintains the majority of BGLs within the target range during the initial phase. However, while both algorithms lead to a decrease of in-target BGLs when the initial BGL was > 130 mg/dL, the IIP does so at a cost of significantly increasing the frequency of hypoglycemia (as high as 50%). No such increase is seen in the PDP.

The NGLI with the IIP is noted to significantly increase in the initial phase as the initial BGL increases. This is a result of the magnitude and slope of the overshooting. No such relationship is seen in the oscillating phase since the trajectories have stabilized. The average NGLI for the IIP in the initial period is 3466 $(\text{mg/dL})^2/\text{hr}^2$. This translates into an average BGL change of 58.8 mg/dL per hour. The maximum NGLI is greater than 21,000 $(\text{mg/dL})^2/\text{hr}^2$ which equates to a BGL change of more than 145 mg/dL per hour. The PDP, however, very effectively reduces the variability seen with the IIP. In the initial phase, the average NGLI is 1117 $(\text{mg/dL})^2/\text{hr}^2$, which corresponds to a BGL change of 33.4 mg/dL per hour (a 43% reduction in variability). The maximum NGLI with the PDP is 8272 $(\text{mg/dL})^2/\text{hr}^2$. This is 63% less than the IIP and equates to a BGL change of 91 mg/dL per hour.

It appears that the risk of hypoglycemia with the IIP is significantly increased by two primary mechanisms. The first involves the "overshooting" phenomenon during the initial phase. Indeed, data in Table 3.2 show that when the initial BGL is high, overshooting causes a significant percent (as high as 20% with a maximum of 50%) of low BGL readings. The second mechanism of hypoglycemia is by wide variation during the oscillating phase. In the oscillating phase, when considering out of target readings, there is a higher rate of hypoglycemia than hyperglycemia (with the maximum of low BGL percentage being above 40%). Thus, for some subjects, the risk of hypoglycemia persists even when the control pattern stabilizes.

Strengths of the IIP during in-silico evaluation include that most BGLs are kept within the target range during the "oscillating" phase as well as during the "initial" phase when the starting BGL is within target range. Weaknesses of the protocol include episodes of severe overshooting and oscillations (sometimes large) in BGL trajectory. These oscillations appear to be the result of the discrete nature of the infusion rules in the IIP. That is, because infusion rates are determined by a limited number of adjustment rules, the insulin bolus amount and infusion rate take jumps as the BGL changes. As a result, the IIP controller may fail to stabilize an individual at the "equilibrium" state and, instead, oscillate between different infusion values. Finally, the IIP (as a result of overshooting and oscillation) is not very effective at reducing hypoglycemia and BGL variability.

Tuning protocol parameters on the T1DMS model represents a challenging optimization problem. As explained before, it is difficult to apply classic linear system control theories to the highly complex nonlinear glucose simulation model. Our strategy in solving this complex non-linear optimization problem is to systematically characterize the impact of parameters on performance metrics by running numerical simulations and finding the optimal trade-off configuration (Figures 3.5 and 3.6). One particular challenge with this is the inter-subject variability. While an insulin insensitive subject may favor a more aggressive protocol, that same protocol may cause significant hypoglycemia in subjects who are highly insulin sensitive. We design the PDP to minimize hypoglycemia over a broad range of glucose trajectories as the optimal configuration must achieve good performance on the entire target patient population.

3.3.5 Formal Verification of the New Protocol

In the previous section, we evaluate the IIP and PDP by closed-loop simulation using the T1DMS together with its virtual subjects. While simulation using high-fidelity models provides vital insights into the physiological impact of a behavior model,

there is no formal guarantee that the virtual subject set of the T1DMS covers the entire Type 1 diabetics population. To this end, formal verification can provide a new level of safety assurance to clinical practitioners before performing human clinical trials.[3]

In this section, we introduce the model of the closed-loop IOGC system as a case study verification benchmark: the model contains both the FDA-accepted high-fidelity physiological model and the PDP that we developed as an enhancement to the IIP. We provide over-approximated value ranges of all model states and parameters. The ranges of the values are extracted from extensive clinical studies [106, 76, 155, 138, 158]. Then, we implement the benchmark in a recently proposed SMT-based hybrid system verification tool, dReal/dReach [97]. Last, we present a proof-of-concept safety verification of the intraoperative glycemic control benchmark over subspaces of physiological parameters and states.

Problem Formulation

In this section, we define the safety verification problem considered in this work. We represent the combined PDP and physiological process (defined in the next section) as a standard hybrid system,

$$\mathcal{H} = \langle \mathcal{X}, \mathcal{Q}, \mathcal{X}_{init}, \mathcal{X}_{inv}, \mathcal{F}(\mathcal{P}), T \rangle \,,$$

where $\mathcal{X}$ represents the continuous states, $\mathcal{Q}$ denotes the discrete modes, $\mathcal{X}_{init} \in \mathcal{R}_{\mathcal{X}}$ specifies the initial condition space, $\mathcal{F}(\mathcal{P})$ captures the flows parameterized by a vector $\mathcal{P} \in \mathcal{R}_P$, $\mathcal{X}_{inv}$ identifies invariants mapping modes to flows, and T relates the transitions between modes. A measurable output $y = \phi(t; \mathcal{X}_{init})$ denotes the glucose value, with $\phi(t, \mathcal{X}_{init})$ describing the measurement at time $t \in [0, t_{\max}]$ [4], having

[3] Currently, model-based trials are only approved to replace pre-clinical testing. It is unclear whether model-based trials will ever be approved to replace clinical (human) testing due to unmodeled physiology and comorbidity inherent in all models.

[4] $t_{\max}$ represents the maximum time the patient is in surgery.

evolved from initial condition $\mathcal{X}_{init}$. We aim to solve the following safety verification problem:

$$\forall t \in [0, t_{\max}], \ \forall \mathcal{P} \in \mathcal{R}_P, \ \forall \mathcal{X}_{init} \in \mathcal{R}_{\mathcal{X}}, \quad y \notin \mathcal{R}_{unsafe},$$

where $\mathcal{R}_{unsafe}$ is a region representing unsafe blood glucose levels (i.e., hypoglycemia and hyperglycemia that are defined clinically [150]).

Modeling the Closed-loop System

The full T1DMS model contains three sub-models (insulin, glucose, and carbohydrate-ingestion) with 13 states and 32 parameters. The original publications [73, 151] discuss the details of physiological modeling and our paper [56] summarizes the model equations from the literature. Since intraoperative patients receive insulin and glucose via intravenous infusion, the two subcutaneous insulin compartment states and the entire carbohydrate-ingestion sub-system can be neglected, resulting in a 7-state intraoperative model, as described in the remainder of this subsection.

The intraoperative model contains an insulin sub-model and a glucose sub-model. The insulin system is a 5-state linear model driven by the insulin input, $u(t)$, written as

$$\dot{I}_p(t) = -(m_2 + m_4)I_p(t) + m_1 I_l(t) + u(t) * 10^2/BW \tag{3.1a}$$

$$\dot{X}(t) = P_{2U}/V_i I_p(t) - P_{2U} X(t) - P_{2U} * I_b \tag{3.1b}$$

$$\dot{I}_1(t) = k_i/V_i I_p(t) - k_i I_1(t) \tag{3.1c}$$

$$\dot{I}_d(t) = k_i I_1(t) - k_i I_d(t) \tag{3.1d}$$

$$\dot{I}_l(t) = m_2 * I_p(t) - (m_1 + m_3) I_l(t). \tag{3.1e}$$

The $I_p(t)$ and $I_l(t)$ states represent insulin mass in the plasma and liver, respectively. $I_1(t)$ and $I_d(t)$ represent a delayed insulin transportation process. $X(t)$ represents an insulin signal in the remote tissue that governs glucose concentration in the interstitial compartment. The model contains a set of parameters that are

patient dependent: $m_{1...4}$ and P_{2u} are rates of insulin mass diffusion among different compartments, V_i is the insulin distribution volume, and BW is the body weight.

The glucose system has two states and is written as

$$\begin{aligned}\dot{G}_p(t) =& - k_1 * G_p(t) + k_2 * G_t(t) - F_{snc} + m(t) * 10^3/BW \\ & + \max\left(0, k_{p1} - k_{p2} * G_p(t) - k_{p3} * I_d(t)\right) \\ & - 1 - \max\left(0, k_{e1} * (G_p(t) - k_{e2})\right)\end{aligned} \tag{3.2a}$$

$$\dot{G}_t(t) = - \frac{(V_{m0} + V_{mx} * X(t)) * G_t(t)}{K_{m0} + G_t(t)} + k_1 * G_p(t) - k_2 * G_t(t) \tag{3.2b}$$

where, $G_p(t)$ and $G_t(t)$ represent the glucose concentration in plasma and interstitial fluids, respectively. The $G_p(t)$ derivative (Equation 3.2a) contains two saturation switches $\max\left(0, k_{p1} - k_{p2} * G_p(t) - k_{p3} * I_d(t)\right)$ and
$\max\left(0, k_{e1} * (G_p - k_{e2})\right)$, which represent the endogenous glucose production (EGP) and renal glucose clearance, respectively. These two max switches yield four discrete modes in the hybrid system representation of the model, and transitions among the four modes are governed by saturations of the two max terms. The G_t derivative contains a non-linear term $-\frac{(V_{m0}+V_{mx}*X(t))*G_t(t)}{K_{m0}+G_t(t)}$ that represents the remote insulin signal $X(t)$'s impact on glucose dynamics. The model contains two population static parameters k_{e1} (glomerular filtration rate) and k_{e2} (renal threshold of glucose). All other parameters are patient dependent: k_1 and k_2 are the glucose exchange rates between the G_p and G_t compartments; k_{p1} is the extrapolated EGP; k_{p2} is the liver glucose effectiveness; k_{p3} is the insulin action on liver; V_{m0}, V_{mx}, and K_{m0} are model parameters that govern the insulin action on G_t; V_g is the glucose distribution volume; $m(t)$ is the intravenous glucose input into the plasma compartment.

The 7-state intraoperative glucose control model is observed through $y(t) = G_p(t)/V_g$, corresponding to the plasma glucose measurement (in mg/dL). Most of the patient-dependent parameters, except for a few such as the body weight, are not measurable in standard hospital tests. Estimating those parameters on individ-

Table 3.5: Over-approximated ranges of the T1DMS model states

States	Ranges	Units	Example Nominal Value
I_p	$[0, 30]$	pmol/kg	5
X'	$[-500, 500]$	pmol/liter	30
I_1	$[0, 300]$	pmol/liter	120
I_d	$[0, 300]$	pmol/liter	120
I_l	$[0, 30]$	pmol/kg	3
G_p	$[0, 1000]$	mg/kg	200
G_t	$[0, 1000]$	mg/kg	150

ual patients involves invasive and costly procedures such as the triple-tracer meal protocol experiment [22, 74], which is clearly not feasible in surgical settings. The FDA-accepted T1DMS simulator comes with 10 adult virtual subjects, each of which is a whole realization of the parameters. Those virtual subjects are extracted from the same distribution as the 100 FDA-accepted adult virtual subjects for black-box controller evaluation were.

All the states and parameters in the FDA-accepted model have physiological meanings, and numerous clinical studies have investigated the ranges of values across different populations [106, 76, 155, 138, 158]. Table 3.5 lists over-approximated ranges and the units of the seven states and Table 3.6 lists over-approximated ranges of the eighteen parameters.

The PDP updates $uc(k)$, $ub(k)$, and $m(k)$ based on $y(k)$ and $y(k-1)$ according to the rules defined in Table 3.7. As described in Section 3.3.4, the PDP's parameters are tuned to minimize the hypoglycemia risk while maximizing quality of glucose control.

Hybrid System Modeling

We model the 7-state intraoperative physiological model and the PDP as a hybrid system as illustrated in Figure 3.7. It is standard practice to perform perioperative monitoring of the patient to ensure the patient is stable enough for surgery [180].

Table 3.6: Over-approximated ranges of the T1DMS model parameters.

Parameters	Ranges	Units	Example Nominal Value
m_1	$[0.1, 1]$	min^{-1}	0.2
m_2	$[0.1, 1]$	min^{-1}	0.3
m_3	$[0.1, 1]$	min^{-1}	0.3
m_4	$[0.05, 0.5]$	min^{-1}	0.1
k_i	$[0.001, 0.02]$	min^{-1}	0.01
P_{2u}	$[0.01, 0.1]$	min^{-1}	0.03
V_i	$[0.02, 0.1]$	liter/kg	0.06
I_b	$[0, 300]$	pmol/liter	100
BW	$[0, 300]$	kg	90
k_1	$[0.02, 0.1]$	min^{-1}	0.05
k_2	$[0.05, 0.3]$	min^{-1}	0.1
k_{p1}	$[1, 10]$	mg/kg/min	5
k_{p2}	$[0.0001, 0.01]$	min^{-1}	0.004
k_{p3}	$[0.001, 0.03]$	mg/kg/min per pmol/liter	0.01
V_{m0}	$[1, 10]$	mg/kg/min	5
V_{mx}	$[0.01, 0.15]$	mg/kg/min per pmol/liter	0.05
K_{m0}	$[100, 1000]$	mg/kg	200
V_g	$[1, 5]$	dL/kg	2

Table 3.7: The PDP rules

Condition	Control Input Update
$y(k) \leq 60$	$uc(k) = 0, ub(k) = 0, m(k) = 12.5$
$60 < y(k) < 100$ AND $y(k) - y(k-1) < -30$	$uc(k) = 0, ub(k) = 0, m(k) = -0.1 * (y(k) - y(k-1))$
$100 \leq y(k) < 300$ OR $y(k) - y(k-1) \geq -30$	$uc(k) = \max(0, 0.05 * (y(k) - 100) + 0.06 * (y(k) - y(k-1))) + 1)$, $ub(k) = 0, m(k) = 0$
$y(k) \geq 300$	$u(k) = 15, ub(k) = 15, m(k) = 0$

During the perioperative period (typically at least 30 minutes), if the patient exhibits extreme glucose variation, the surgery may be postponed until the patient stabilizes. To model the perioperative monitoring procedure, we divide the verification time into two phases: During the initial monitoring phase, if the glucose output y leaves a control range (e.g., $70 - 130$ mg/dL), the system transitions into the "NOT ADMIT" mode; if the glucose output y stays within the control range during the entire monitoring period, then the system transitions into the protocol control phase and

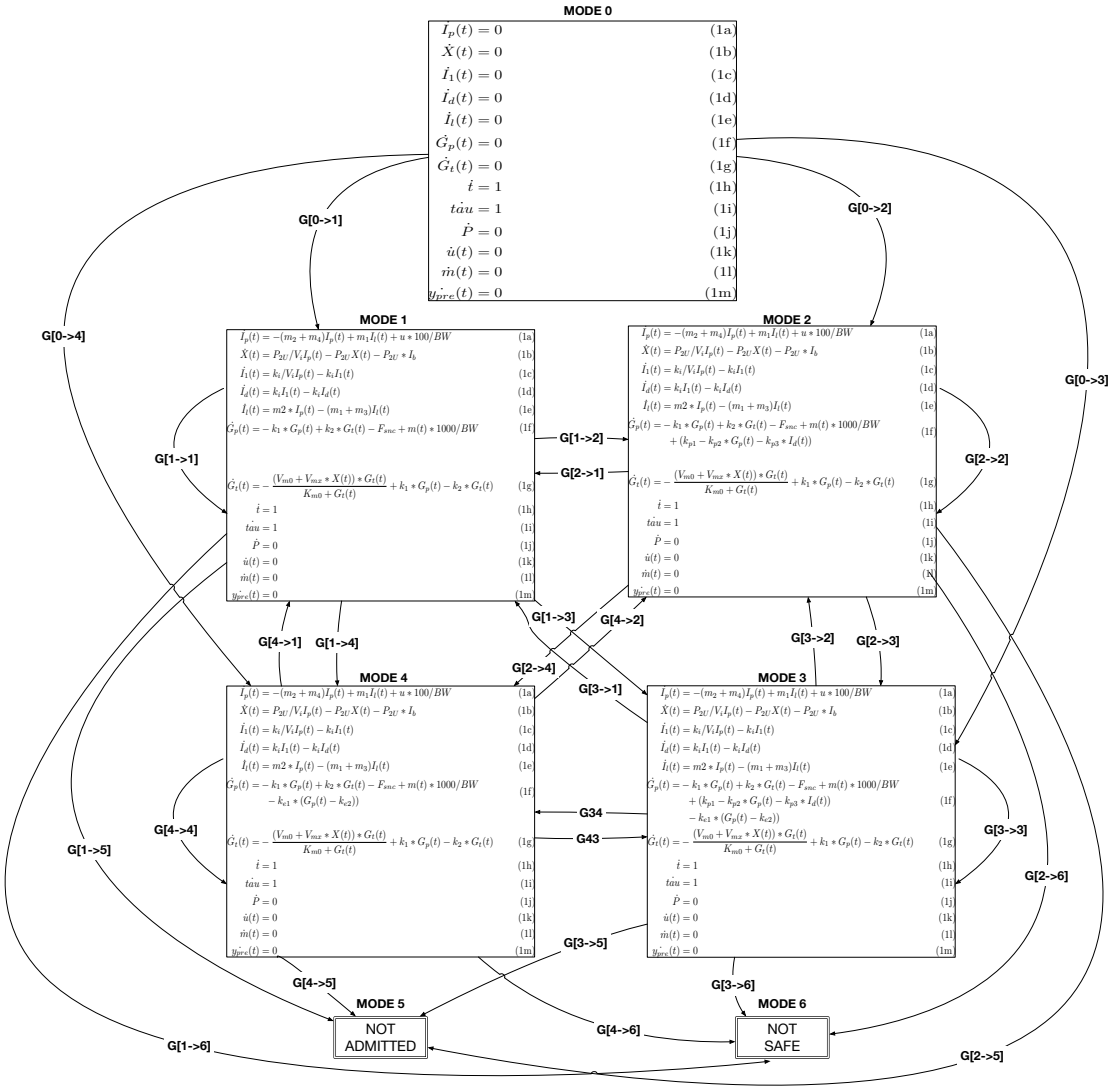


Figure 3.7: A hybrid system representation of the FDA-accepted high-fidelity physiological model with the PDP.

the PDP starts operating. During the protocol control phase, the system transitions into the "NOT SAFE" mode if the glucose output y leaves a safe range (e.g., $60-150$ mg/dL).

The hybrid system contains seven states: one initial state mode 0; four states (modes 1 - 4) that represent the system dynamics with four possible combinations of the two saturation switch terms in Equation 3.2a, which are re-stated in Equation 3.3;

one "NOT ADMIT" mode and one "NOT SAFE" mode.

$$\begin{aligned} &\max(0, C_1), \ \textit{where } C_1 = k_{p1} - k_{p2} * G_p - k_{p3} * I_d \\ &\max(0, C_2), \ \textit{where } C_2 = k_{e1} * (G_p - k_{e2}) \end{aligned} \tag{3.3}$$

The system has 30 continuous states[5]

$$\mathcal{X} = \{I_p, X, I_1, I_d, I_l, G_p, G_t, \mathbf{P}, t, tau, y_{pre}, u, m\},$$

where $\mathbf{P}$ denotes the 18 model parameters, t is the global verification time, tau is the local timer variable, $y_{pre}(t)$ is a variable to record the last output sample, $u(t)$ and $m(t)$ are the insulin and meal inputs.

For simplicity of presentation we denote the four combinations of the two max terms using T_1 to T_4, as shown in Equation 3.4.

$$\begin{aligned} T_1 &:= (C_1 \leq 0) \wedge (C_2 \leq 0) \\ T_2 &:= (C_1 > 0) \wedge (C_2 \leq 0) \\ T_3 &:= (C_1 > 0) \wedge (C_2 > 0) \\ T_4 &:= (C_1 \leq 0) \wedge (C_2 > 0) \end{aligned} \tag{3.4}$$

Mode 0 is the initial state, in which all states have zero derivatives except t and tau. The system immediately goes into one of modes 1 - 4. The invariant on mode 0 is $INV_0 := (tau \leq 0)$. Equation 3.5 defines the guards on the transitions out of mode 0.

$$\forall i \in \{1, 2, 3, 4\}, \ G[0 \rightarrow i] := T_i \wedge (tau \geq 0) \tag{3.5}$$

Let $t \in [0, t_a]$ denote the monitoring phase. Let $\mathcal{R}_{na}$ and $\mathcal{R}_{unsafe}$ denote the set of "NOT ADMIT" glucose values and "NOT SAFE" glucose values, respectively. Equation 3.6 defines the invariants on modes 1 - 4. To model the practical scenario

[5]To be consistent with the dReach implementation explained next, in the hybrid system model, we denote all parameters as continuous states with derivatives of zero (i.e., constants).

that a clinician may not check exactly at the 30 minutes mark, we allow timing non-determinism by relaxing the conditions on the invariants with a sampling jitter δ.

$$\begin{aligned} \forall i \in \{1,2,3,4\}, INV_i :=& (\neg(t \leq t_a \wedge y \in \mathcal{R}_{na}) \\ & \wedge (\neg(t > t_a \wedge y \in \mathcal{R}_{unsafe}) \\ & \wedge T_i \\ & \wedge (tau \leq 30 + \delta)) \end{aligned} \tag{3.6}$$

The self-transitions on modes 1 - 4 are triggered at the glucose sample times. On the self-transitions $\forall i \in \{1,2,3,4\},\ G[i \rightarrow i]$, control inputs u and m are updated according to the PDP, and y_{pre} is updated to the current y. Considering the timing jitter δ, Equation 3.7 defines the self-transition guards.

$$\forall i \in \{1,2,3,4\},\ G[i \rightarrow i] := (tau \geq 30 - \delta) \tag{3.7}$$

The transition guards between modes 1 - 4 are governed by conditions T_1 - T_4 and are defined in Equation 3.8.

$$\forall i, j \in \{1,2,3,4\},\ G[i \rightarrow j] := T_j \tag{3.8}$$

In modes 1 - 4, if $y \in \mathcal{R}_{na}$ during the monitoring phase, the system transitions into the "NOT ADMIT" mode 5. Equation 3.9 defines the transition guards between modes 1 - 4 and the "NOT ADMIT" mode 5.

$$\forall i \in \{1,2,3,4\},\ G[i \rightarrow 5] := (t \leq t_a \wedge y \in \mathcal{R}_{na}) \tag{3.9}$$

In modes 1 - 4, if $y \in \mathcal{R}_{unsafe}$ after the monitoring phase, the system transitions into the "NOT SAFE" mode 6. Equation 3.10 defines the transition guards between

modes 1 - 4 and the "NOT SAFE" mode 6.

$$\forall i \in \{1,2,3,4\},\ G[i \to 6] := (t > t_a \wedge y \in \mathcal{R}_{unsafe}) \tag{3.10}$$

The "NOT ADMIT" mode 5 and "NOT SAFE" mode 6 are terminating states with no invariants or transitions out of them. The safety verification question is specified as follows: For all initial conditions (where the 7 physiological states and 18 parameters are in their ranges), determine if the system can reach the "NOT SAFE" mode 6.

Verification in dReach

The dReach verification tool [149] utilizes the framework of δ-complete decision procedures that aims to solve first-order logic formula with arbitrary computable real functions [98]. The dReach tool can be employed to prove safety properties of hybrid systems over finite time by identifying safe and unsafe regions of the state space and defining a corresponding δ-decision problem. Following [98], we consider the δ-*decision* problem

$$\begin{aligned} &\exists \mathcal{X}_{init} \wedge \exists t \in [0, t_{max}] \wedge \exists y \in \mathcal{R}_{unsafe} \text{s.t.} \\ &\qquad |\mathcal{X}_{init}| \leq \delta_1 \wedge |y - \phi(t; \mathcal{X}_{init})| \leq \delta_2 \end{aligned} \tag{3.11}$$

where δ_i is a numerical error bound specified by an arbitrary rational number and the bounded first-order sentences contain Type 2 computable functions [144].

In this work, we define an unsafe region via limits on the glucose levels observable in the patient. We seek to show that for the PDP, composed with the physiological model described by a hybrid system with non-linear ODEs, there does not exist an initial condition which can lead to the satisfiability of (3.11) within a bounded time. As a conservative solution, the dReach tool (through δ-*weakening*) verifies for all initial conditions and bounded time that either the unsafe region is unreachable

(UNSAT) or the unsafe region is reachable within a δ error (δ-SAT).

The dReach implementation of the surgical glucose hybrid system contains 30 state variables: 7 physiological states; 18 parameters; 2 inputs (insulin rate u and glucose rate m); 1 state to record the last glucose reading; 1 global time state, and 1 local timer state. The dReach source code of this implementation is available online[6].

To perform verification, we employ dReach version 2.15.01 on a Linux server with an Intel(R) Xeon(R) E5-2667 v2 3.30GHz CPU and 64 GB memory, and the results are provided in Table 3.8. First, we note that dReach is a bounded model checker, therefore the search depth or *Path Length* refers to the number of discrete transitions for which we perform verification. In the results, the *Path Length* is the search depth completed by dReach in *Time* concluding in *Result*, where DNF translates to *did not finish* and $\mathbf{x}_0$ and $\mathbf{p}_0$ denote the nominal states and parameters specified in Table 3.5 and Table 3.6, respectively. From the results, we observe that allowing the parameters and initial state to vary simultaneously over their maximum over-approximated ranges prevents dReach from reaching a depth of more than 3. In this scenario, a path length of 3 corresponds to a maximum of one hour of surgery. The fact that dReach can not exceed the arguably trivial depth of 3 after 30 hours suggests that verification over the entire over-approximated parameter and initial condition space is a computationally challenging problem.

To investigate the capabilities of dReach, we allow the initial state to vary over the full range, but constrain the parameters to equal $\mathbf{p}_0$. These results are consistent with the T1DMS scenario for virtual subjects with unknown initial states. Here we observe a significant improvement in verification results, with dReach achieving a depth of 7 in 16.4 hours corresponding to a maximum surgery duration of 3.5 hours. By constraining the initial variance of the state and parameters to a hypercube around the nominal virtual subject, we observe that dReach is able to achieve a depth of 7 in 8.1 hours, corresponding to a maximum surgery duration of 3.5 hours.

[6]URL:https://github.com/chen333/igc-benchmark

Table 3.8: Verification results for $\mathcal{R}_{safe} = [60, 180]$.

Physiological Range		Path	Time	Result
State	Parameter	Length	(hours)	
Full	Full	3	30	safe
Full	Full	4	DNF	-
Full	$\mathbf{p}_0$	3	0.1	safe
Full	$\mathbf{p}_0$	4	0.6	safe
Full	$\mathbf{p}_0$	5	3.1	safe
Full	$\mathbf{p}_0$	6	8.2	safe
Full	$\mathbf{p}_0$	7	16.4	safe
Full	$\mathbf{p}_0$	8	DNF	-
$\mathbf{x}_0 \pm 0.5$	$\mathbf{p}_0 \pm 0.5$	3	0.1	safe
$\mathbf{x}_0 \pm 0.5$	$\mathbf{p}_0 \pm 0.5$	4	0.4	safe
$\mathbf{x}_0 \pm 0.5$	$\mathbf{p}_0 \pm 0.5$	5	1.1	safe
$\mathbf{x}_0 \pm 0.5$	$\mathbf{p}_0 \pm 0.5$	6	2.9	safe
$\mathbf{x}_0 \pm 0.5$	$\mathbf{p}_0 \pm 0.5$	7	8.1	safe
$\mathbf{x}_0 \pm 0.5$	$\mathbf{p}_0 \pm 0.5$	8	DNF	-

This suggests that dividing the parameter and initial condition space can significantly improve time-to-verification given sufficient computing resources.

3.3.6 Towards Run-time Safety Monitoring

In the previous section, we formally verify that a protocol is safe despite uncertain physiological parameters and states that are drawn from continuous sub-regions of the entire physiologically possible variance ranges. Our work provides a stronger safety guarantee than previous studies on evaluating insulin protocols using simulations [170, 182, 283, 164, 277, 178], which can only cover finite samples of uncertain parameters. It is worth noting that within each dReach verification run, the physiological model parameters do not change as time progresses. This assumption is also commonly adopted by most previous simulation-based studies, i.e., once a virtual subject is chosen, it does not change over the simulation run. However, such assumption may be violated in reality, especially in surgical glucose control scenarios. For example, patients can suffer from stress-induced hyperglycemia during surgeries,

which can lead to elevated infection risk [194, 235, 137]. From a modeling perspective, those transient changes in the glucose physiology can manifest as temporal fluctuations in certain physiological parameters, e.g., insulin sensitivity. Coping with such short-term changes of physiological parameters in surgical glucose control is an especially challenging problem considering that most parameters can not be directly measured, and there currently lacks quantitative clinical understanding on which parameters may change and how much they may vary.

In this section, we present a run-time safety monitoring technique that leverages the maximal model to track transient changes in patient's physiology and predict safety-critical events by online re-training of a virtual subject set using real-time glucose measurements. Specifically, in this case study, we aim to predict when glucose levels deviate from a specified region in a near future time window. An implementation of the proposed methodology is evaluated on retrospective real patient data, and the results illustrate that our prediction algorithm achieves 96% sensitivity with an average false alarm rate of 0.5 false alarm per surgery. This technique may complement the verification of protocol-guide control behaviors by providing predictive warnings of critical events (e.g., hypoglycemia) to the human operators.

Overview of the Methodology

Our framework consists of two major steps. First, we generate a covering set (CS) that consists of a large number of computational virtual subjects (CVS). We call the virtual subjects distributed with the T1DM simulator physiological virtual subjects (PVS) because their parameters are derived from experimental physiological data [188]. We validate that the CS can produce glucose value predictions that cover a large range of possible values with a certain degree of uniformity. In the second step, the CS is used in a data-driven adaptive safety monitor that assess the safety of the control input (insulin dosage) suggested by the normal controller. Specifically, we train the CS to learn the real patient's dynamics: we collect a few

past BGL readings from the patient, simulate all CVS in CS on the maximal model, and sort the CVS by comparing the predicted BGL trajectories with the real BGL values. The CS, in which CVS are sorted by prediction errors, are called the trained CS. We then use the trained CS to predict the range of the next BGL, assuming the suggested control input is given. An alarm will be raised if the predicted range of the next BGL overlaps with unsafe BGL region, which implies the suggested control input is unsafe. We evaluate our methodology by replaying the algorithm on retrospective glucose data collected from 51 Type 1 diabetic ICU patients, and for each BGL value y, the algorithm predicts y one check interval ahead and uses the predicted range to classify y as safe (negatives) or unsafe (positives).

Problem Description

Figure 3.8 shows the overall system architecture. The BGL sensor readings are passed to both normal controllers (ICU protocol or other control algorithm implemented by caregivers) and the safety monitor. The normal controller suggests an insulin dosage based on certain algorithms. The safety monitor (details are explained in subsequent sections) predicts the range of the next BGL reading, assuming the suggested insulin dosage is given. If the predicted range is safe, then the suggested dosage is passed to the actuator (pumps). Otherwise, the safety monitor raises an alarm of possible unsafe insulin dosage and feed the information back to caregivers for re-assessment.

We consider the same surgical glucose control scenario described in Section 3.3. Our technique utilize the maximal glucose model which is detailed in Section 3.3.5. In the rest of this section, we use X to denote the 7-dimensional state vector $X = [I_p, X', I_1, I_d, I_l, G_p, G_t]$, y and u to denote the model output and control input, respectively, and P to denote the parameter vector (see Table 3.6 for the parameter meanings and ranges of values). The complete model can be written in an abstract form $\dot{X} = f(X, P, u), y = X(6)/V_g$. A "virtual subject" is a configuration of P.

The safety property comes from the clinical requirement, i.e., the BGL should

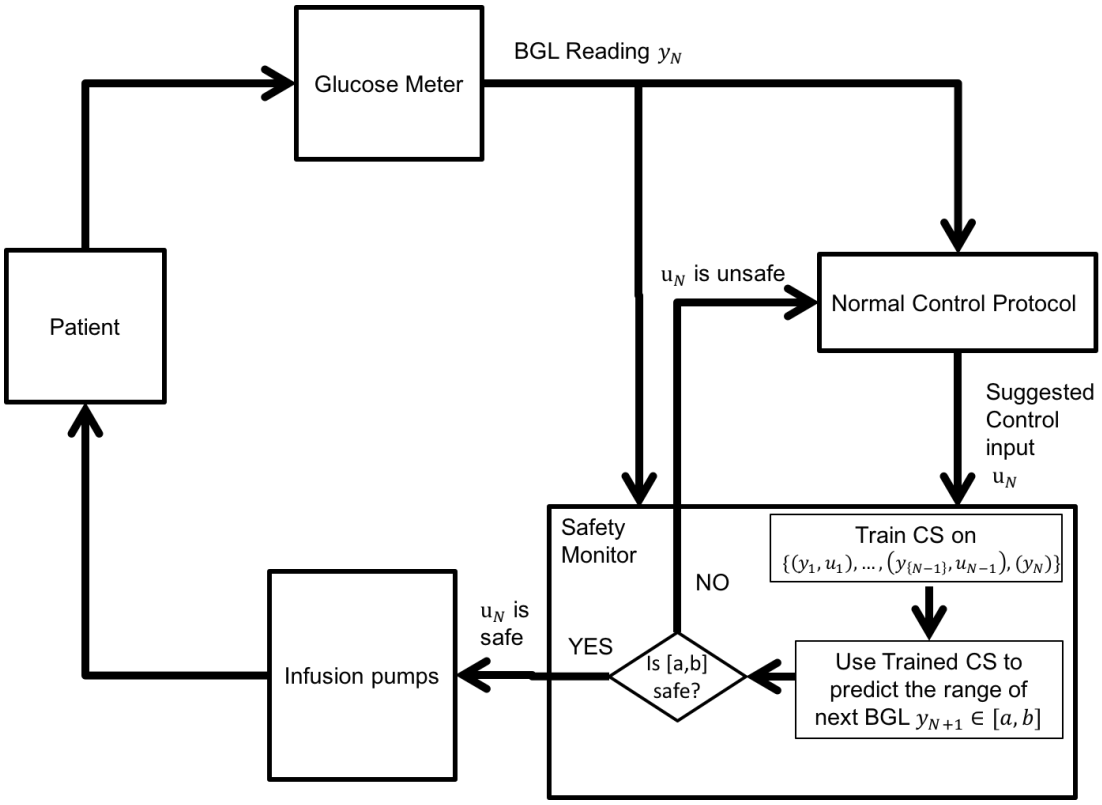


Figure 3.8: Architecture of Safety Monitor for Surgical Glucose Control.

not drop below a critical limit L (e.g., 80 mg/dL) at any time. This is one of the most important safety requirements for BGL regulation [70].

The problem we are solving can be formulated as follows: for an individual patient, at any check point N, given the past sequences of BGL measurements and insulin rates $y_1, \ldots, y_{N-1}$ and $u_1, \ldots, u_{N-1}$, current BGL y_N, and suggested control input u_N, is u_N safe for the patient, that is, is it possible that $y_{N+1} < L$ if u_N is given?

Safety Monitor Design

In this section we present our safety monitor algorithm. The algorithm consists of two phases.

1. In Phase A, we generate a set of CVS (called the covering set (CS)) by randomly

sampling parameter vectors from the bounding hypercube and we show that the CS gives predictions of BGL that cover a large possible range with certain degree of uniformity.

2. In Phase B, we use the CS for safety monitoring. At each check point, we let the CS simulate a patient's past BGL sequence and compute the prediction error of each CVS. The CVS in the CS are then ordered by how well their predictions match the true past BGL. Then we take the suggested control input and use the sorted CVS to predict the range of the patient's next BGL. If the predicted range is unsafe, then an alarm is raised to notify the normal controller and/or the caregiver; otherwise, the control input is passed to the actuator.

In the rest of this section, we explain the technical details of the two phases in our algorithm.

Covering set generation and validation. As mentioned before, one fundamental challenge of using maximal models in glucose control is that most elements of the parameter vector cannot be identified given the currently available clinical data. Here is how we approach the challenge. Suppose the patient is at a certain state (X_0, y_0) and has a true parameter vector P_T. We know nothing yet about P_T except that P_T is bounded in a 20-dimension hypercube H_P. But we know that each vector $P_x \in H_P$ will "drive" the patient $\dot{X} = f(X, P, u)$ to a (probably distinct) ending state (X_{1x}, y_{1x}) after one interval T assuming u_0 is given during the interval. The first question we consider is: if we only know that P_T is somewhere in H_P, what is the distribution of $\{y_{1x}\}$ and which vectors P_x will drive y_{1x} to a specific range?

This question has an intuitive interpretation in glucose control: assuming the patient's current BGL is at y_0 and he/she is given a certain amount of insulin u_0, what is the BGL after a time interval T? In fact, this is the simple procedure that can be used to roughly estimate a person's insulin sensitivity level.

The second question we consider is as follows: Suppose that we can compute the distribution of $\{y_{1x}\}$, assuming a certain starting state (X_0, y_0), and we can pick a set of vectors P_{CS} to uniformly "cover" the P_{CS} is called the covering set (CS) and the vectors in it are computational virtual subjects (CVS), by which we highlight the fact that the CVS are used to generate a computational coverage and distinguish them from PVS. The first question is a CS generation problem and the second one is a CS validation problem.

We cannot use the existing PVS in the T1DM simulator as the CS for our algorithm for two reasons. First, the size of the PVS set is quite limited (300), and we need a much larger set of virtual subjects to generate a good covering range of BGL, which is essential for the safety monitor. Second, the PVS were derived from experimental physiological data such that they can mimic real human patients and can be used in the T1DM simulator to rule out unsafe or ineffective controllers before actual clinical trails. The PVS set is not generated for the computational coverage purpose that we are interested in.

The details of the Phase A are explained as follows.

A.1 CS generation

- **A.1.1** Choose an initial condition (X_0, y_0). The quality of CS will be related to how (X_0, y_0) is chosen. A difficulty here is that the model is unobservable, i.e., for a given y_0, there are infinitely many corresponding X_0's. Inspired by the idea that an "insulin sensitivity test" tries to stabilize the patient's physiological states before the test starts, we tackle the difficulty in the following way: the academic version of T1DM Simulator includes 10 PVS that are drawn in the same way as the FDA-approved PVS population, we run the 10 PVS on the simulator and obtained the initial states generated by the simulator. To fully excite the dynamics and explore a large range of $\{y_{1x}\}$, we choose a high initial BGL $y_0 = 250$ mg/dL and let u be the "250 mg/dL BGL" action item defined in the previously mentioned HUP IIP, which is 10 U/hr infusion

for $T = 30$ minutes and 10 U insulin bolus given immediately. We experiment with 10 PVS' X_0's established by the T1DM simulator and pick the one that gives the largest distribution range of $\{y_{1x}\}$.

In general hospital ICUs, it is not possible to monitor X_0 from real patients. What we do here is let the T1DM Simulator (together with its PVS) start at a realistically high BGL, give a real hospital protocol-defined insulin dosage, and establish the X_0, which is the best we can do given the technologies and data available. The quality of CS will be validated and further evaluated in the subsequent steps.

- **A.1.2** Given the (X_0, y_0, u_0) and T, we randomly sample P_{CS} from the bounding hypercube H_P[7], simulate the model on each vector $P_x \in CS$, and get the corresponding y_{1x} after T. We collect a large set (1.5 million in our implementation) of sample vectors P_x so that the distribution of $\{y_{1x}\}$ covers a sufficiently large range. The sufficiency can be justified by common clinical knowledge of how fast human BGL can drop: the BGL decline/rise rate is subject to physiological limits, e.g., it is written in HUP Glycemic Control Protocol for Cardiac Surgical Patients that "the average rate of decline should be no more than 50 mg/dL per hour" (so clearly 50 mg/dL per hour is considered as a high BGL decline rate that should be avoided). In our algorithm the $\{y_{1x}\}$ actually covers a range that is several times larger than $\pm$ 50 mg/dL per hour.

- **A.1.3** From the large set of randomly sampled $\{P_x\}$ and the corresponding $\{y_{1x}\}$, we select a subset of $\{P_x\}$ to form the CS P_{CS}, such that the corresponding $\{y_{1CS}\}$ uniformly cover the entire range of $\{y_{1x}\}$. The CS is not constrained by experimental data and can be much larger (in the implementation we have 10,000 CVS in the CS) than the FDA-approved PVS population

[7]Due to the lack of accessible clinical knowledge of how the physiological parameters are correlated statistically, the sampling process does not assume any prior correlation between different parameters. One of the current limitations of this work is the lack of statistical guarantee that the sampled virtual population sufficiently represent all real patients

(300).

A.2 CS validation

In Step A.1.3, the CS is selected to uniformly cover $\{y_{1x}\}$ given the (X_0, y_0, u_0) picked in Step A.1.1. We can not simply conjecture that CS will generate a uniformly distributed predictions of y_{N+1} for any starting state and input tuple (X_N, y_N, u_N). Therefore we need to validate CS for different tuples (X_N, y_N, u_N) and test the coverage of predicted $\{y_{N+1}\}$. The validation algorithm works as follows:

- **A.2.1** First, we generate a set of test cases to validate the CS. Each test case is a four-value tuple (X_N, y_N, u_N, y_{N+1}). Ideally, such test cases should be from real clinical data, but as mentioned before, it is impossible to monitor X_N directly from patients. So instead, we run a real hospital protocol (the HUP IIP) on the T1DM simulator (with its 10 PVS) at different initial conditions, and we obtain a large set of simulated patient BGL trajectories, which are as closest to be real as we can get.

- **A.2.2** We then test the entire CS on every single test case obtained above. Specifically, for each test case (X_N, y_N, u_N, y_{N+1}), we simulate the model on every $P_x \in P_{CS}$ for one interval, starting from (X_N, y_N) and using u_N as the control input. At the end of the interval, we get a y_x which is the prediction of the true value y_{N+1} by a CVS in CS. We then measure the quality of coverage of all the predictions $\{y_x\}$ by two requirements: 1) $\{y_x\}$ should contain y_{N+1}, i.e., $y_{N+1} \in [\min\{y_x\}, \max\{y_x\}]$; 2) $\{y_x\}$ is distributed around y_{N+1} with a certain degree of uniformity, which is precisely defined in the next step.

- **A.2.3** In Step A.1.3, the CS is selected such that the $\{y_{1x}\}$ are perfectly uniformly distributed. However, due to the non-linear nature of the model, it should not be expected that the same CS will generate $\{y_x\}$ that have the same perfect uniform distribution starting from any initial state (X_N, y_N, u_N, y_{N+1}). In addition, for the safe control purpose, we do not need a perfectly uniform

distribution. Instead, what we need in the data-driven adaptive safe control step is that there are enough candidate predictions in $\{y_x\}$ that fill a neighboring region of y_{N+1}. Therefore, for the control purpose, a less than perfect uniform distribution is good enough.

The uniformity metric we use to test the coverage of $\{y_x\}$ is therefore defined as follows:

- We first determine the neighboring region of y_{N+1}. Again, the size of the region depends on the size of the maximum prediction range in the control step, which is set to 60 mg/dL. So here we look at the $[\max(y_{N+1} - 30, 30), y_{N+1} + 30]$ neighboring region. The lower bound saturates at 30 because $BGL < 30$ mg/dL is extremely low BGL which happen with very low probability not only in practice but also in simulation, so we should not expect a lot of predictions go below 30 mg/dL. In addition, 30 mg/dL is much lower than our safe limit $L = 80$ mg/dL, so there is a wide $30 - 80$ "buffer zone" for the safety monitor to raise an alarm.
- Finally, we test the minimum density of predicted BGL values in $\{y_x\}$ that fall into the neighboring region. The density is defined as the average number of values in $\{y_x\}$ that fall into a unit length (1 mg/dL) BGL interval. The density is computed by a binning algorithm: put the values in $\{y_x\}$ that are in the neighboring region into small-sized (5 mg/dL) bins and find the minimum counts bin to calculate the minimum density. If the "density" is no less than 1 counts per mg/dL, i.e., for every possible integer BGL readings there are at least one prediction, then the CS passes the coverage testing on case (X_N, y_N, u_N, y_{N+1}). For extra redundancy, our algorithm actually achieves minimum density of 8 counts per mg/dL.

Data-driven adaptive safe monitoring. The Phase B is repeated at each check point. We use the identified and validated CS to predict a patient's BGL one

check point ahead, given only the past BGL readings and insulin inputs. This is, in general, very challenging, especially when using the maximal model. Existing model-predictive control (MPC) approaches for glucose control either use a simple linear model to approximate the non-linear dynamics, or require costly parameter pre-tuning. It has been pointed out that there is a fundamental analytical limitation of parameter identification on high-dimension unobservable non-linear models; that is, it is not possible to uniquely identify so many unknown parameters given only the limited single input and single output data [63].

Instead of trying to directly identify P_T, we propose a data-driven technique to adaptively train CS on past sequence (y, u) (the past sequence gets updated each step as new measurements and control actions are taken), and then use the trained CS to predict the range of next BGL reading. The predicted range is then used for the safety monitoring.

B.1 CS Training: The training set is a sequence of past BGL and control actions, $\{(y_i, u_i), \ldots, (y_{N-1}, u_{N-1}), (y_N)\}$. N is the current check point and y_N is the current BGL. The training set contains the latest $N - i + 1$ BGL and $N - i$ control actions (a control action is effective for an interval between two y's so there is one less control action than the number of y's). Initially $i = 1$. These records are accessible in ICUs, e.g., at HUP, the BGL readings, control information, and patient-related information are recorded electronically. The CS is trained as follows:

B.1.1 We simulate the model on the CS, starting from y_i. Each $P_x \in P_{CS}$ will generate a corresponding simulated trace $\{y_{ix}, \ldots, y_{Nx}\}$.

B.1.2 The simulated traces are compared to the true trace $\{y_i, \ldots, y_N\}$, and L-2 norm errors are calculated for each trace.

B.1.3 We sort the CS by the non-decreasing order of the prediction errors and the sorted CS is called the trained CS.

B.2 Range prediction: To predict the range of the next BGL y_{N+1}, we initialize a list R_{N+1} as empty.

B.2.1 We start from the top of the trained CS, retrieve a vector P_x, extend the simulation $y_{ix}, \ldots, y_{Nx}$ by one step (assuming the suggested control input u_N is given at check point N), put the predicted $y_{(N+1)x}$ into R_{N+1}, move onto the next vector in the trained CS, and repeat the process.

B.2.2 The list R_{N+1} holds predictions of y_{N+1} of a top subset of CVS in the trained CS. The minimum and maximal values of R_{N+1} are the predicted range for y_{N+1}.

We repeat filling predictions $y_{x(N+1)}$ into R_{N+1} until at least one of the following two conditions becomes true:

- The predicted range exceeds a pre-defined window size. This window size directly affects the R_{N+1} and the performance of the prediction algorithm. We develop a double-zone strategy to determine it.

  If $\min(R_{N+1})$ is above some threshold W_b, then the stop condition is

  $$(\max(R_{N+1}) - \min(R_{N+1})) > W_H,$$

  where W_H is the maximum window size of the "high" zone. If

  $$\min(R_{N+1}) < W_b,$$

  then the stop condition is

  $$(\max(R_{N+1}) - \min(R_{N+1})) > W_L,$$

  where W_L is the maximum window size of the "low" zone.

  The idea is that when the predicted BGL in R_{N+1} are relatively high (above W_b), we allow a larger prediction window. And when some predicted BGL in R_{N+1} are in the low zone, we narrow down the prediction window because now

the predicted BGL are closer to the unsafe region and a narrower window will help reduce the false positive rate.

- The bottom of the trained CS is reached; i.e., all predictions by the CS have been put into R_{N+1}).

The predicted range of y_{N+1} is given by $[\min(R_{N+1}), \max(R_{N+1})]$.

B.3 Robust safety monitoring: Using the predicted range

$$[\min(R_{N+1}), \max(R_{N+1})],$$

if $\min(T_{N+1})$ is less than the pre-defined safety limit L, then it implies the suggested control input u_N can drive y_{N+1} into the unsafe region, and an alarm is to be raised and fed back to caregivers. Otherwise, u_N is granted to the actuator. This is currently a YES/NO classification alarm, and one can also use $[\min(T_{N+1}), \max(T_{N+1})]$ to generate a more informative, fuzzy logic type alarm: for example, having different levels of urgency depending on how much $[\min(T_{N+1}), \max(T_{N+1})]$ intersects with the unsafe region.

B.4 Adaptive training set adjustment: The training sequence grows as more BGL readings are collected. Sometimes the real BGL trajectory could exhibit "turns" that cannot be predicted by models.[8] An interesting phenomenon is that the unmodeled dynamics can not only cause prediction errors at the "turns", but also affect subsequent predictions after the "turns" even when the patient's physiological

[8]In ICU surgical patient data, we have seen some BGL changes that cannot be well explained by models: for example, when the BGL tends to stabilize around a certain level and insulin infusion rate does not change for a while, there are sometimes sudden BGL increases, i.e., the patient appears to be more resistant to insulin for a short period. Anesthesiologists at HUP are interested in such scenario when we replayed the retrospect BGL data and specifically pointed out a few such "turns". Their medical opinions are that there are some other factors, e.g., body temperature and surgery-induced stress level, that they believe also change patients' insulin sensitivity, but those effects are not modeled by even the state-of-the-art maximal models. The doctors think it would be useful if we can compare the model-predicted BGL with the true BGL in real time and alert them when such "turns" happen (which can be done with our framework) as they believe these events could suggest some physiological state changes that they concern.

states go back to the baseline. This is because in the learning step, the CS is trained by prediction errors calculated from the entire training set. If there is are unpredictable "turns" in the set and we keep it for future predictions, then essentially the CS will always try to learn the "turns" because they dominate the prediction error.

To cope with such cascading error issue, we dynamically adjust the training sequence in real time. Whenever a true reading y_{N+1} lands outside the predicted range, we remove all past readings before y_N and the new training sequence starts from y_N for future predictions (we need to keep y_N because we need at least one interval, i.e., two past readings, for training when predicting y_{N+2}). The rationality is that a BGL "turn" indicates a possible change the patient's physiological parameters, so the safety monitor should also be reset to match such change, instead of keeping old past readings and still trying to learn the "turns" starting from an old initial states.

Evaluation

We implement the complete safety monitoring algorithm and evaluate it on de-identified, retrospective patient glucose data collected from the Hospital of the University of Pennsylvania (with the Institutional Review Board (IRB) approval). In this section, we first present the results of the implementation and the evaluation using real patient data. Then we discuss several design trade-offs regarding how to configure our algorithm according to different clinical requirements and needs.

CS generation and validation. Starting from $y_0 = 250$ mg/dL, $u_0 = 10$ U/hr infusion plus 10 U insulin bolus and setting the sampling interval T to be 30 minutes, we explore the distribution of predicted $\{y_1\}$ given randomly sampled CVS $P \in H_P$. We aim at generating 10,000 CVS for the CS, and to fully explore the possible distribution of $\{y_1\}$, we sample $1,500,000$ CVS in H_P and the distribution of $\{y_1\}$

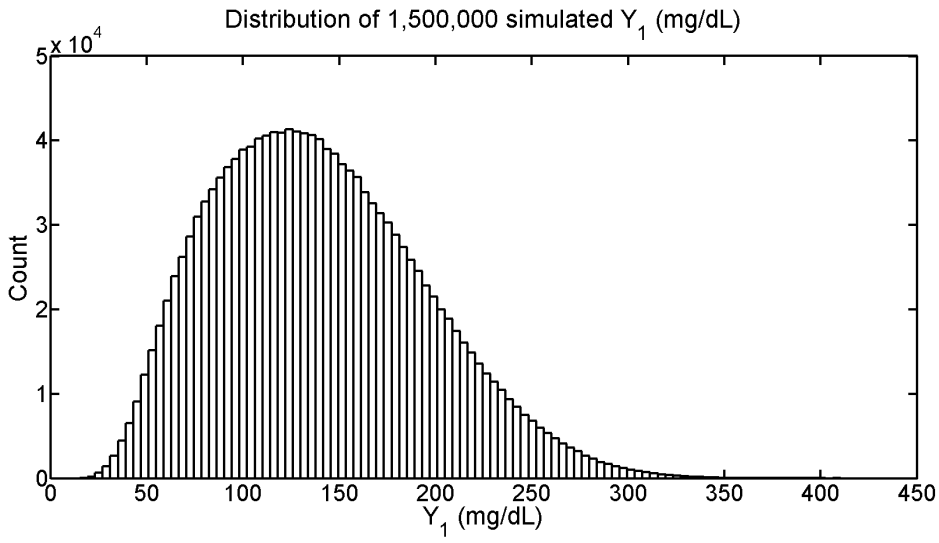


Figure 3.9: Distribution of $1,500,000$ simulated y_1.

is shown in Figure 3.9.

The distribution covers a large range from 20 mg/dL to as high as 400 mg/dL. Hospital protocols consider 50 mg/dL per hour a dangerously high BGL decline rate. The lowest predicted BGL after 30 minutes is 20 mg/dL, which translates into a 460 mg/dL per hour drop rate, more than 9 times larger than what the protocols consider dangerous. On the highest end, the highest predicted BGL is around 400 mg/dL, but it is medicine common sense that 10 U/hr is a fairly high insulin dosage, and it is very unlikely that a patient's BGL can even increase from 250 mg/dL to 400 mg/dL in 30 minutes under such high insulin rate. Therefore, it is justified that the simulated $\{y_1\}$ covers a sufficiently large range. From the 1.5 million candidate CVS, we select 10,000 CVS into the CS such that the $\{y_1\}$ coverage of the CS is uniformly distributed, as shown in Figure 3.10.

To validate the coverage of $\{y_{(N+1)x}\}$ produced by the CS given any starting state, we extract test cases from the simulated BGL trajectories that are obtained by running the T1DM Simulator simulator together with the HUP IIP protocol controller at different initial conditions. We simulate the 10 PVS included in the T1DM simulator starting from 18 different initial BGL (from 70 mg/dL to 240

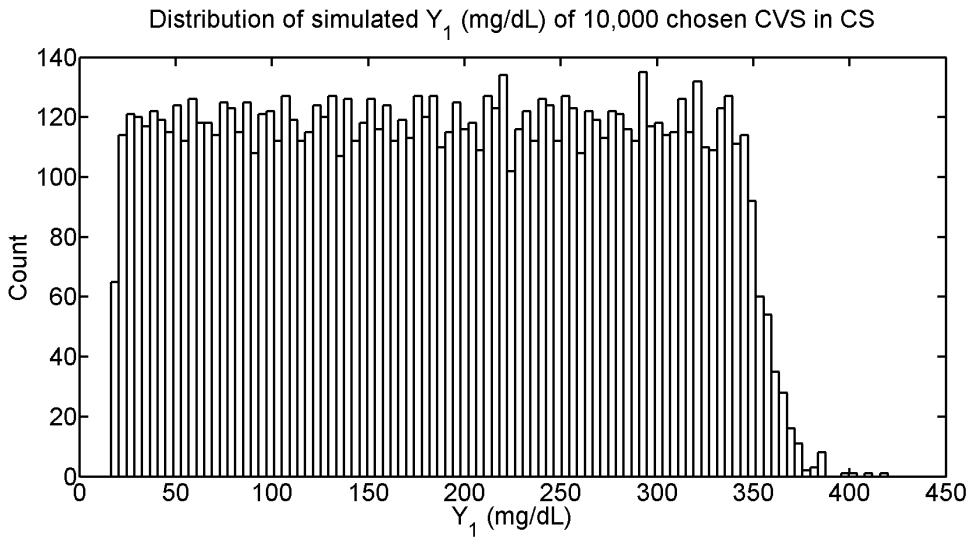


Figure 3.10: Distribution of the simulated y_1 of the 10,000 CVS chosen by CS.

mg/dL, 10 mg/dL step increase) and obtain 180 simulated BGL trajectories. A test case is extracted at each 30 minutes check point of a trajectory, and for each trajectory we extract test cases from the first 24 check points (time 0 to 12 hour), because after 12 hours the simulated BGL trajectories are oscillating around an equilibrium (the initial transient response fades away), so the states simply repeat in a periodic pattern. Therefore we get 4320 (10*18*24) test cases. For each test case, we calculate the minimum density of CVS in the neighboring window of y_{N+1}.

Figure 3.11 shows the distribution of the minimum density values of CVS in all test cases. The overall minimum density value in all 4320 test cases is 8.6 counts per mg/dL, which is greater than the required 1 per mg/dL, i.e., the generated CS passes all 4320 coverage tests.

Safety monitor evaluation. We use the generated CS to test our safety monitor. For safety and regulatory reasons, we cannot directly test this newly designed monitor technique on human patients without extensive offline experiments. So we demonstrate the validity of our design by replaying retrospective patient BGL data on the safety monitor, which, from a computational perspective, is the same as if

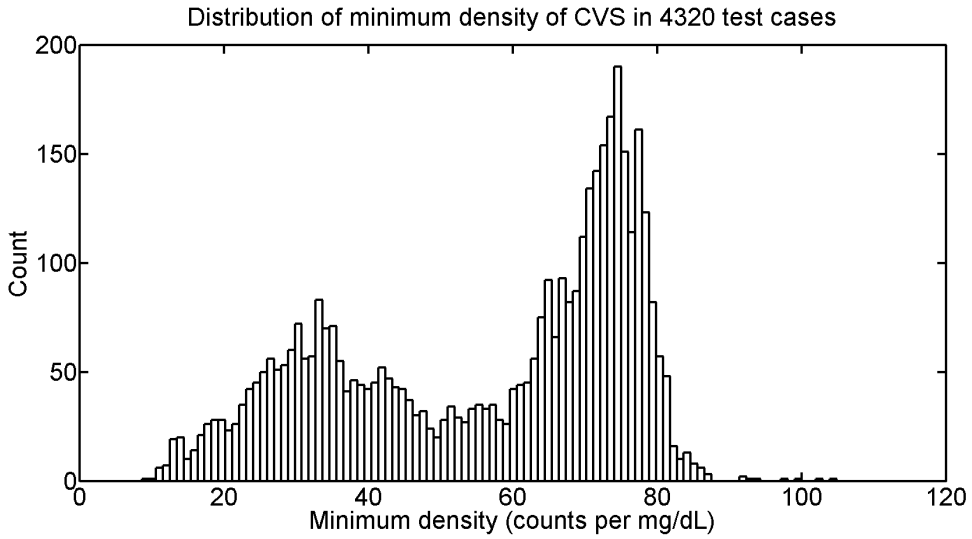
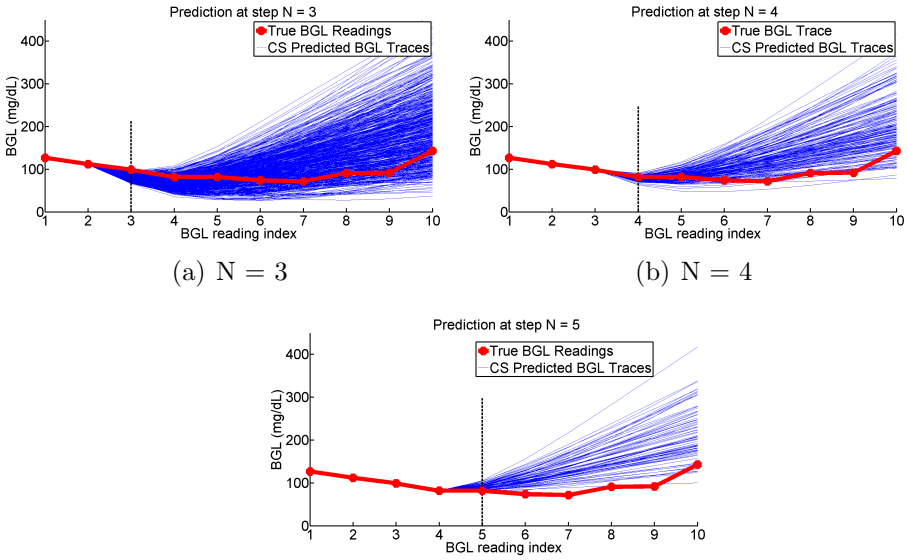


Figure 3.11: Distribution of minimum density of CVS in 4320 test cases.

the safety monitor is tested in the real clinical environment. The data is collected from 51 de-identified (to protect privacy) Type 1 diabetic patients that received cardiac surgery and were controlled by the HUP IIP (so insulin inputs are known). As defined by the protocol, the BGL data were taken every 30 minutes.

The evaluation algorithm works as follows. We retroactively run the HUP IIP (as the normal controller) and our safety monitor on the real BGL data. At each real BGL reading, the safety monitor computes the range of the next BGL reading given the control input determined by the HUP IIP and predicts whether or not the next BGL reading will be safe. Then we move on to the next BGL reading, check the value to verify if the prediction made in the last step is correct, and repeat the process.

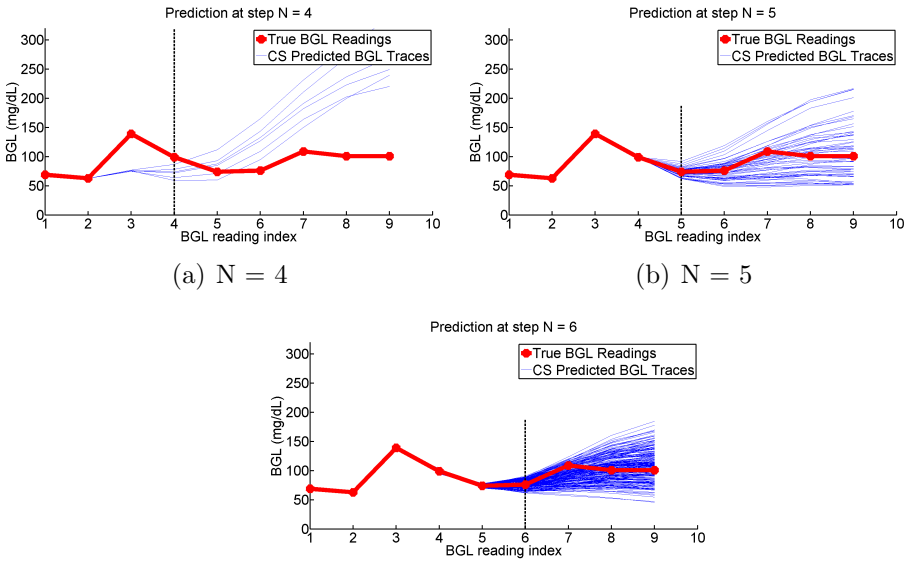
CS training and range prediction. For each patient, we start prediction on y_3 (y_1 and y_2 are used as the initial training set) and move forward until the end of each data trace. Figure 3.12 illustrates how the adaptive learning algorithm works on a patient's data trace (it is case No.1). At each check point, the algorithm predicts y_{N+1} one interval ahead by picking those CVS that achieve lowest prediction errors



(a) N = 3 (b) N = 4

(c) N = 5

Figure 3.12: Prediction snapshots at $N = 3, 4, 5$ for patient No.1.



(a) N = 4 (b) N = 5

(c) N = 6

Figure 3.13: Illustration of adaptive training set adjustment on case No.2.

on the past sequence $(y_i, \ldots, y_N)$. As shown in Figure 3.12, our algorithm adaptively tracks the true BGL trend. At $N = 3$ there are only 2 history readings and 1 interval in the training set. The training set is so small at that point that it cannot fully

Table 3.9: Evaluation results on 51 patients' data (144 prediction points) when L is set to 100 mg/dL

L = 100		
	Safe (True)	Unsafe (True)
Safe (Predicted)	95	1
UnSafe (Predicted)	24	24

separate the large CS. That is why the predicted values of y_3 cluster around the true y_3 but the BGL trajectories are further apart in the future. But the algorithm only needs to look one step ahead at a time, so future divergences are irrelevant (as Figure 3.12 shows, the predicted range converges as the algorithm moves forward to $N = 4$ and 5).

We test our algorithm on 51 patients' trajectories, the lengths of which vary depending on how long the surgeries were. Overall there are 246 BGL readings, 195 BGL intervals, and 144 BGL readings for the algorithm to predict (the first two readings of each patient are needed for initial training). The performance of the safety monitoring algorithm can be tuned by setting the threshold L and window sizes differently. In practice, those parameters should be set according to caregivers' clinical needs. According to the IIP, clinicians consider BG less than 60 mg/dL as critical condition and start to take precautions when BG is within the $60-99$ mg/dL region. Therefore, a reasonable setting of alarming threshold L would be 100 mg/dL, so that caregivers can receive predictive alerts on risky BG trends. Table 3.9 reports the performance matrices of the algorithm when L is set to 100 mg/dL (the window size settings are as follows: $W_b = 110$, $W_H = 60$, and $W_L = 30$). The result shows a 96% sensitivity (24 out of 25 unsafe events are correctly identified) with less than 0.5 false alarms per operation period on average (24 false alarms on 51 patients and each patient's data is collected from one operation period).

3.4 Summary of this Chapter

In this chapter, we have proposed a model-based framework to analyze and assure the safety of generic (i.e., non-personalized) user behaviors in medical CPS, which are typically guided by rule-based protocols. By applying the framework to an intraoperative glycemic control case study, we have identified limitations of a current clinical protocol (the IIP) and designed a new protocol (the PDP) to overcome its weaknesses while preserving its strengths. We formally verified that the new protocol ensures safety for a virtual population of an FDA-accepted physiological model that is instantiated with uncertain initial physiological states.

Existing related work in model-based evaluation of protocols has predominantly relied on Monte Carlo simulation, and the key limitation is that, given the complexities of the physiological models, there is no guarantee the discrete sample set of unknown parameters/states cover all clinical scenarios. Our work has demonstrated that it is possible to leverage hybrid system model checking to realize a new level of safety guarantees in evaluating protocol-driven behaviors: We formally verified that a protocol ensures safety for a set of virtual subjects that map to continuous subspaces of uncertain parameters/states, i.e., the protocol is proven robust to uncertainties. To the best of our knowledge, this is the first work towards formally verifying an insulin protocol using the most advanced maximal non-linear glucose physiological model that contains numerous unidentifiable parameters and unobservable states. Our verification results also revealed that allowing all 18 parameters and 7 states of the non-linear physiological model to simultaneously vary within their respective over-approximated ranges poses a computationally challenging problem to a state-of-the-art hybrid system model checker. Therefore, we have presented this problem (through our publication [55]) to the hybrid system community as a benchmark, which represents a clinically important problem, for evaluating and improving verification tools.

To cope with the practical challenge that a patient's physiological parameters

may exhibit transient fluctuations in reality, we have developed a run-time safety monitoring technique to adaptively track the physiological changes using the maximal model and provide caregivers a predictively alarm on critical events. We applied the technique to the intraoperative glucose control case study and developed a novel computational virtual subject (CVS) based adaptive technique for robust safety monitoring. Preliminary evaluation results using a clinical dataset shows the proposed safety monitor achieves high sensitivity with a low false alarm rate.

Chapter 4

Model-Based Analysis of Personalized Behaviors

In Chapter 3, we develop a modeling paradigm for generic (i.e., non-personalized) behaviors that are driven by rule-based protocols, which are common in hospital care. In this chapter, we consider personalized user behaviors. This type of behaviors is frequently observed in out-patient care scenarios such as home care and mobile health, in which case the users can exercise a high degree of discretion in how they want to use with the medical CPS according to their individual preferences and habits.

Part of the work described in this chapter has been published in our previous paper [54].[9]

The rest of this chapter is organized as follows: Section 4.1 motivates the problem; Section 4.2 proposes a methodological framework that enables systematic identification of behavior variables and instantiating the behavior models by leveraging domain knowledge and clinical data; Section 4.3 applies the proposed approach to modeling individualized insulin pump user behaviors; Section 4.4 concludes our work

[9]The publisher and/or the copyright agreement grant using any portion of the paper in a dissertation.

in this research thrust.

4.1 Problem Description

In recent years, the healthcare industry has observed a rapidly growing class of patient-centered medical technology that aims at constantly monitoring and improving an individual's health conditions by real-time physiological sensing and therapy delivering [42, 113, 134]. Examples include body area networks [133], mobile health [167], and chronicle disease management [177]. Those applications are mostly designed for out-patient use scenarios, where users are usually patients themselves and may exhibit distinct behaviors in their interaction with the systems. There is a critical need to quantitatively model personalized user behaviors in this new type of medical systems [156]. Modeling individual user's behaviors has at least two major benefits. First, behavior models represent higher-level knowledge learned from the raw data, which can help clinicians and patients more efficiently apprehend health conditions and diagnose potential problems. There is a fundamental gap in the vast amount of health information available on individual patients and the very limited per-patient access to healthcare professionals. For example, Type 1 diabetics who use insulin pumps have their glucose levels measured every five minutes, yet a patient typically sees a clinician for a short routine visit every few months. It is clearly infeasible for a clinician to manually sift through the vast amount of raw data points trying to uncover relevant trends during a half-hour visit. Second, behavior models are critical to evaluating the safety of new systems that aim at achieving higher autonomy. Traditional medical devices for out-patient care are relatively low-risk monitoring devices such as vital sign monitors for home use. Recent ubiquitous healthcare applications start to include treatment devices, thereby closing the loop with promoted levels of automation, e.g., the semi-autonomous glucose management systems for diabetics [66]. Such systems involve potentially life-threatening risks

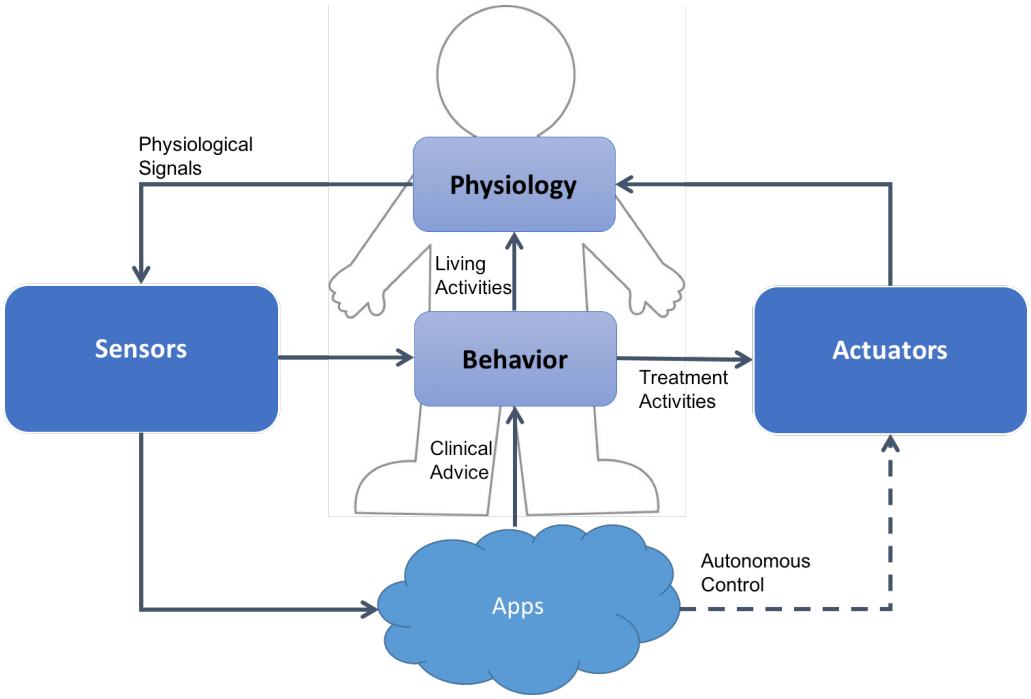


Figure 4.1: An architecture of patient-centered healthcare applications.

(e.g., insulin overdose) and therefore demand new safety analysis techniques that explicitly consider user behaviors.

We present a generalized architecture, as shown in Figure 4.1, that represents a wide range of patient-centered healthcare applications, e.g, wearable body sensors [181] and the artificial pancreas [197]. The sensors send physiological measurements to the patient and the Apps. The "Apps" component represents decision support applications that may run on local and/or remote computing platforms. The Apps may also serve as an interface to remote healthcare consultation with clinicians [134]. If autonomous control is allowed, the Apps may directly send commands to actuation devices (e.g., infusion pumps). The user monitors physiological measurements from the sensors and receives decision support feedback from the Apps. The patient impact his/her own physiology through two types of activities: living activities such as eating and exercise, and treatment activities such as medication intake. Those two types of activities are represented in Figure 4.1 as two actuation

channels from behavior to physiology: Living activities directly impact physiology; Treatment activities need to be actuated by medical devices, which broadly represent any devices or agents (e.g., medications) that are involved in the treatments.

One unique feature of patient-centered medical applications, compared to other human-in-the-loop systems such as vehicles and robots, is the direct action channel from behaviors to the physiology. In other application domains, e.g., automobiles and robotics, operator behaviors can only change the state of the physical process (e.g., vehicle dynamics or robot movements) through mechanical or electrical "actuators", i.e., the behavior impact is constrained by the design of the actuators. In medical systems where users are patients themselves, the users can access more "control surfaces": In addition to controlling the actuators such as infusion pumps, users can directly influence their own physiology by behaviors such as eating and exercising, which are not constrained by actuators. Therefore, it is especially important to understand the dynamics and impact of behaviors in this type of medical CPS, because it is impossible to guarantee safety solely by system design without considering how users would behave.

We consider the behavior modeling problem within the context of patient-centered healthcare applications, as illustrated in Figure 4.1. The goal is to model how behaviors (e.g., the living and treatment activities) are driven by the information that users receive/perceive (e.g., physiological measurements and clinical advice).

The exact forms of behavior models clearly depend on applications. Our contribution to this problem is two folds. First, we propose a framework to systematically identify and model relevant behavior factors in applications that share the architecture depicted in Figure 4.1. Second, we apply the modeling framework to a concrete case study in Section 4.3, and we demonstrate how to instantiate a concrete behavior model by leveraging both domain knowledge and clinical data.

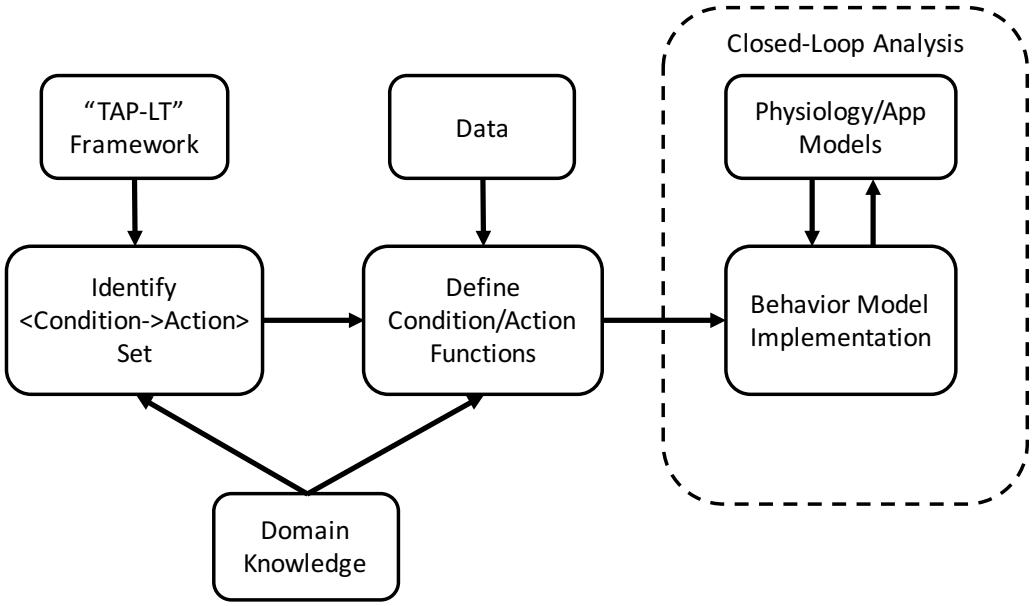


Figure 4.2: A methodological framework for analyzing personalized behaviors.

4.2 The Behavior Modeling Framework

We propose a behavior modeling framework, as shown in Figure 4.2, that consists of several stages. First, we define behaviors as a set of "conditions-actions" pairings. We introduce the "Time-Apps-Physiology triggered Living-Treatment actions" (TAP-LT) framework to systematically identify relevant behavior variables. Next, we instantiate the behavior model by incorporating clinical data and domain knowledge. In the last step, we implement the behavior model in modeling tools and compose it with the physiology & App models for closed-loop analysis. In the rest of this section, we describe the key components of this framework, and in Section 4.3, we apply this framework to an insulin pump case study.

4.2.1 Identify Behavior Variables

Engineering psychology researchers decompose the human decision making into several key stages including perception, cognition, and response [273]. In the patient-centered healthcare applications, the user receives information from sensors and

Apps, and he/she impacts physiology through various activities. We define behaviors as a set of condition-action mappings [$Condition_i \rightarrow Action_i$], denoting that if $Condition_i$ is true, the user will take $Action_i$.

We identify three types of conditions and two types of actions by analyzing the general architecture in Figure 4.1. Inspired by the concepts of time-triggered and event-triggered actions, which are widely adopted in many engineering systems (e.g., real-time scheduling and distributed control) [7], we categorize the triggering conditions of a user's actions into time-driven and event-driven conditions. Within the healthcare application architecture, the event-driven conditions can be further divided into two types: events generated by the physiology and events generated by the Apps. Therefore, there are three types of conditions: *Time*-driven, *Apps*-driven, and *Physiology*-driven conditions (denoted as TAP conditions). On the action end, there are two types of actions that correspond to the two activity channels, as discussed previously: *Living* activities and *Treatment* activities (denoted as LT actions).

A user's actions may be triggered by one or a combination of the three conditions in the TAP-LT framework. For example, eating regular meals is a time-driven living activity, and taking medications following Apps' advice is an Apps-driven treatment activity. A diabetic patient regularly checking blood glucose levels and only injecting insulin if the glucose level is too high is an example of an action (insulin injection) triggered by a combination of time-driven conditions (regular checkpoints) and physiology-driven conditions (high glucose level).

4.2.2 Formulate Behavior Functions

After identifying the relevant behavior condition and action variables, the next step is to define the functional mapping between conditions and actions. Each condition-action pair is formulated as $\langle \mathbf{h}(t, y_P, y_A) = True \rightarrow [u_L, u_T] = \mathbf{g}(t, y_P, y_A, u_L, u_T) \rangle$, where t denotes time, y_P denotes physiological information, y_A denotes Apps' feed-

back, u_L and u_T denote living and treatment actions, respectively[10]. As an example, consider a diabetic patient checking the glucose level (y_{BG}) every 120 minutes, and if the glucose level is higher than 180 mg/dL, he/she takes an insulin dose u_T calculated as $(y_{BG} - 180)/20$ (the ratio 20 is called insulin sensitivity, a metric for tuning insulin doses based on y_{BG}). This specific behavior can be formulated as $\langle (mod(t, 120) = 0 \wedge y_{BG} > 180) = True \rightarrow [u_L, u_T] = [0, (y_{BG} - 180)/20] \rangle$.

Formulating the condition and action functions typically requires integrating qualitative understanding from domain knowledge and quantitative analysis of patient data. For example, in the glucose control application, from general clinical knowledge, we know that patients tend to take correction insulin when the glucose level is high. However, the frequencies of glucose checking and correction may greatly vary across individuals, and therefore the personalized behavior model needs to be inferred by analyzing data.

4.2.3 Closed-Loop Safety Analysis

The behavior model, which consists of $[Condition \rightarrow Action]$ relations, is implemented in modeling tools for closed-loop analysis. Selecting the appropriate modeling and analysis tool would depend on how the condition and action functions are formulated, e.g., whether the model is continuous time or discrete time and whether the behavior functions are deterministic or probabilistic. In the next section, we apply the general framework to modeling insulin pump users' behaviors, and we express the behavior model in a formalism that allows probabilistic formal verification of the behaviors' impact on physiology.

[10]Here we follow the control system notation convention, in which y represents measurements and u represents control inputs to the control plant (the physiological process in our case).

4.3 An Insulin Pump Therapy Case Study

In this section, we apply the behavior modeling framework to an out-patient diabetes care application. This section is organized as follows: Section 4.3.1 motivates the problem; Section 4.3.2 summarizes the contributions of this case study; Section 4.3.3 presents the "Eat, Trust, and Correct" (ETC) behavior analysis, which is an application of the proposed TAP-LT framework; Section 4.3.4 introduces a data-driven technique to quantitatively model ETC behaviors; Section 4.3.5 describes probabilistic formal verification of the behavior model using an individualized physiological model.

4.3.1 Motivation

Diabetes affects approximately 29 million people (or 9.3% of the population) in the United States and is the seventh leading cause of death [95]. Type 1 diabetics (more than 1 million in the United States) and some Type 2 diabetics depend on intensive daily insulin therapy to control their blood glucose level and to avoid numerous serious long-term complications of hyperglycemia, such as cardiovascular disease, nerve damage, blindness, and kidney damage. Advanced insulin pump technology provides continuous subcutaneous insulin infusion (CSII) therapy. It is estimated that about $400,000$ T1D patients in the United States use insulin pumps [28]. Reviews of clinical studies suggest that CSII provides improved glycemic control [200, 25].

Current insulin pumps require close supervision from the user in many operational aspects. The user needs to do a carb count for every meal so that the pump software can recommend an insulin bolus dose based on the estimated carbohydrate ratio and insulin sensitivity parameters. The user needs to approve or modify every software-recommended bolus dose. Due to safety concerns, there is currently no insulin pump approved to the U.S. market that can deliver boluses automatically without user acknowledgment.

A recent official consensus statement by clinical expert committees stresses critical needs on evidence-based research to better understand the impact of insulin pumps on diabetic users in various physiological, psychological, and social aspects (see the AACE/ACE report [103]). Clinical studies on the use of insulin pumps predominantly focus on evaluating the impact on physiological metrics, such as the mean glucose value, rate of hypoglycemia, and HbA1c levels [128, 91, 227]. Very few results exist on understanding the behavioral aspects of how diabetic users interact with insulin pumps, which are important factors in assessing how much a patient may benefit from the CSII therapy [103]. For example, the behavioral factors include the user's eating patterns, adherence to pump-recommended insulin doses, and the level of attention to glycemic control.

Recent advances in insulin pump technology demonstrate a clear trend towards a high level of automation [2]. At the same time, a proven safe and effective fully closed-loop glycemic control system that requires no user supervision is not likely to be available in the near future [197]. The emerging smart insulin pumps [285] introduce new challenging engineering concerns: For example, how much the user will trust the automation features, whether he/she will eat more carbohydrates while believing the pump's safety features can "handle it", and whether he/she may become less attentive to glucose monitoring. One of our previous papers discusses the potential hazards associated with the shared human-software control in a multi-mode artificial pancreas system [222].

4.3.2 Contributions

We develop a novel data-driven technique to instantiate the TAP-LT based behavior model in the insulin pump application. The instantiated ETC model enables analyzing common behavior patterns within the patient population. For effective clustering, we design a technique to reduce the dimensionality of the model to a compact representation that retains most of the information based on a key observa-

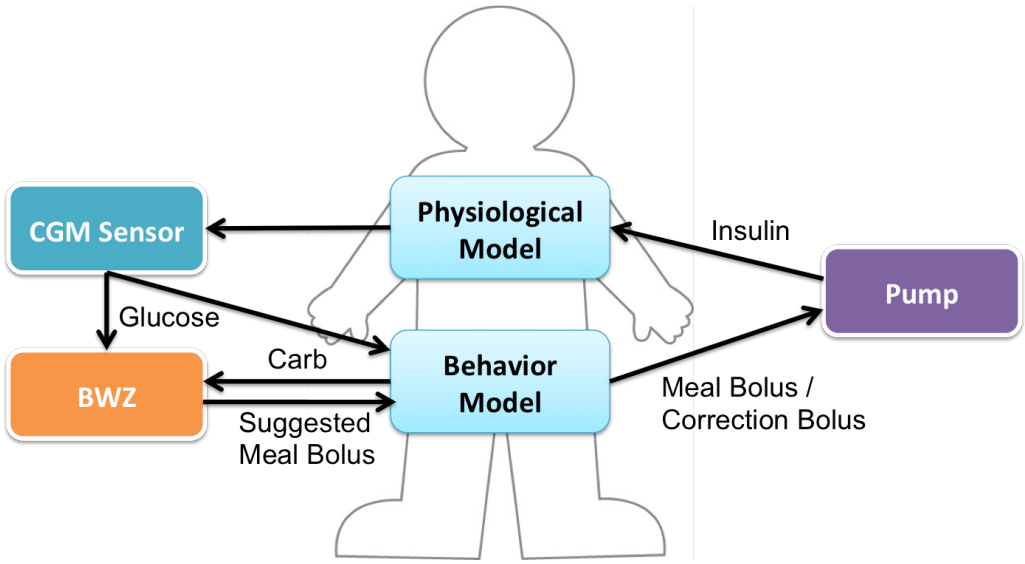


Figure 4.3: The CSII system architecture.

tion about the model. We validate the dimension reduction and clustering method by cross validation. The ETC model reveals new quantifiable behavior patterns in clinical data, which enable personalized clinical diagnosis and patient education, as confirmed by medical expert review of the results. Formal verification of the behavior model demonstrates that switching behaviors may improve individual clinical outcomes. The verification results provide quantitative evidence of how diabetic patients may achieve better glycemic control by behavior changes.

4.3.3 "Eat, Trust, and Correct" (ETC) Behavior Model

In this section, we introduce an "Eat, Trust, and Correct" (ETC) modeling framework to analyze the behavior patterns of T1D insulin pump users. Figure 4.3 illustrates an overview of the CSII system. In this system, the user and patient are the same person. The Continuous Glucose Monitoring (CGM) sensor periodically transmits a subcutaneous glucose measurement to the pump and the CGM readings can also be seen by the user. When the user eats, he/she inputs the meal information

into the pump Bolus Wizard® (BWZ) software[11], which is a bolus advisory feature that calculates a recommended insulin bolus dose. The user needs to approve or modify every BWZ-recommended bolus. The user can also initiate correction insulin boluses at any time. In current insulin pumps, the BWZ cannot deliver insulin without the user's approval.

By applying the TAP-LT framework to analyzing the CSII system, we identify that the user exercises control authority through three channels: 1) eating, a living activity that represents the user's internal interaction with his/her own physiology; 2) approving or modifying BWZ-recommended boluses, which is an App-triggered treatment activity; 3) taking correction insulin boluses, which is a time- and physiology-triggered treatment activity. Based on this observation, we propose the ETC behavior modeling framework that includes the three key components of the user's behaviors in the CSII system:

- **Eat**: How often a patient eats throughout a day and what the meal carb count distributions are at different times of the day;
- **Trust**: The likelihood of a patient following the BWZ recommended bolus doses, and if not, how much dosage he/she adjusts;
- **Correct**: How often a patient takes correction boluses and what the dose distributions are at different times of the day.

In this study, we aim at modeling quantifiable behavioral metrics from available clinical data. The ETC behavior model represents the statistical trends in observable user activities of the insulin pump application. We use the terms Eat, Trust, and Correct as concise references to the three behavioral aspects in the case study.

[11] All patients in this study use the Medtronic systems, in which the bolus recommendation software is called Bolus Wizard (BWZ). In this section, we use "BWZ" to generally refer to the bolus advisory feature. The methods and findings are certainly not specific to any particular manufacturer's products.

Although the modeling technique is applicable to other MCPS systems, the meaning of the ETC model is specific to its application context. For example, the Trust component of the ETC model captures a pattern in the data that indicates user adherence to software recommendations, but the model is not intended to describe users' inherent psychological trust level of an MCPS system in general.

We collect the CSII system data from 68 T1D patients during their clinical visits to the diabetes center in the University of Pennsylvania Health System (with IRB approval). The patients all use insulin pumps augmented with a CGM sensor. The average time range of a patient's data is 35 days. A national registry of T1D patients receiving care in diabetes centers, of which Penn is a participating center, indicates that 60% of adult patients use insulin pumps, and 15% use CGM sensors [23]. So from the 932 patients with T1D seen at the University of Pennsylvania in the past year, 84 would be expected to use both an insulin pump and a CGM sensor. Thus, the 68 patients included in this study represent the majority of patients expected to be utilizing this sensor-augmented CSII technology in the management of their T1D at the University of Pennsylvania Health System.[12]

Thirteen of the 68 patients have data that are from different continuous time periods (we call a continuous period a "segment" in the rest of this section). For each of those patients, the data segments are typically separated by several months. Other patients have one data segment each. All together there are 92 data segments from the 68 patients, with an average segment duration of 26 days. Since there is no reason to assume the same patient's behavior does not change from one segment to another, we use the segment (referred to as "patient-segment" in the rest of this section) as the unit entity when analyzing behaviors. [13]

The dataset includes two parts: 5-minute sampled CGM measurements and insulin pump data. The insulin pump data contains two sections: insulin delivery logs

[12]We had to exclude some patients from the dataset because of missing data, i.e., for those patients, there are not enough CGM measurements, and insulin pump records that overlap in time.

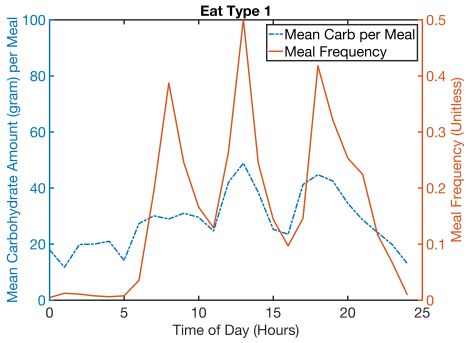
[13]In the behavior analysis, we actually identify a few patients whose behaviors change between segments.

and BWZ data. The insulin delivery logs record the insulin basal rate at points of change, the user selected insulin bolus doses, and the pump delivered insulin bolus doses. The insulin basal rate is a low continuous infusion rate and it changes at several pre-scheduled times of the day. The insulin boluses consist of mealtime boluses and non-mealtime boluses, which we call correction boluses. All data are time-stamped to the precision of second.

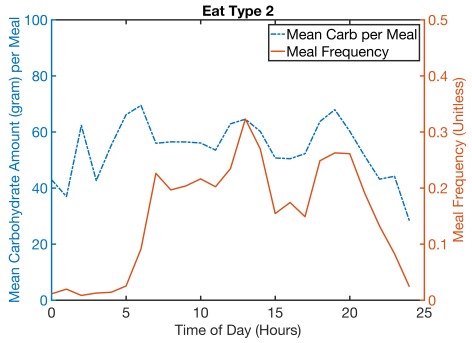
The BWZ calculates recommended bolus doses based on three pieces of information: (1) the meal bolus dose that is calculated from the carbohydrate input and the estimated patient-specific carbohydrate ratio, which represents the insulin dose needed for each unit carbohydrate input; (2) the correction bolus dose that is calculated from the difference between the current glucose level and the target glucose level (e.g., 100 mg/dL), and the estimated patient-specific insulin sensitivity, which represents the insulin dose needed for lowering a unit glucose value; (3) the active insulin on board, which is an estimated amount of residual insulin in the physiological system. The BWZ data section includes the following data fields: user-reported carb counts, estimated correction bolus doses, estimated meal bolus doses, estimated active insulin on board, the target glucose levels, carbohydrates ratios, insulin sensitivity values, and BWZ-recommended bolus doses. Next, we present the key findings of analyzing the ETC behavioral metrics from the CSII clinical dataset.

"Eat" Behavior Analysis

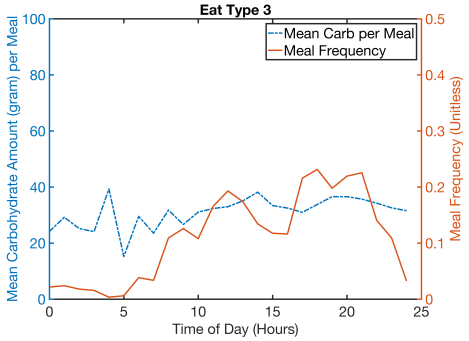
The BWZ data contains patient-reported meal carb counts and mealtimes. For each patient, we aggregate the meal data of each patient-segment and calculate the per-segment distribution of the carb count at different times of the day. We then feed all patient-segments' meal distribution data into clustering algorithm to identify common patterns within the population (technical details of the clustering method are explained in Section 4.3.4). Each cluster contains a subset of the patient-segments with similar meal distribution patterns. We identify three Eat clusters from the CSII



(a) Cluster E1



(b) Cluster E2



(c) Cluster E3

Figure 4.4: Aggregated mean daily meal intake distributions of all patient-segments in three Eat clusters. E1 shows three prominent peak mealtimes with a low likelihood of carb intake between regular meals. E2 shows regular peak mealtimes with an elevated likelihood of carb intake between regular meals. E3 shows no regular peak mealtimes, and carb intake spread throughout a day.

dataset. Figure 4.4 illustrates the aggregated average daily meal intake statistics of the patient-segments within each cluster.

To calculate meal frequency over the data collection time interval, we round each mealtime to the nearest whole hour time. In Figure 4.4, the red solid line (mapped to the right Y-axis) is a connected-dot line of meal frequencies at 25 hour times (0-24 Hours): For example, a meal frequency of 0.4 at time 8 AM in Figure 4.4(a) means for those patients in cluster E1, on average 40% of the days a patient would eat a meal around 8 AM. The corresponding point on the blue dashed line (mapped to the left Y-axis) is the mean carb count per meal over all the meals around 8 AM of

all the patients within cluster E1.

Figure 4.4 shows three distinct daily meal intake patterns, denoted as clusters E1, E2, and E3. The E1 cluster shown in Figure 4.4(a) represents patients who consistently eat three regular meals (peak frequency times are breakfast around 8 AM, lunch around 1 PM, and dinner around 7 PM) with some morning and afternoon snacks around 10 AM and 4 PM, respectively. The E1 patients rarely eat in the late night or early morning. The E2 cluster shown in Figure 4.4(b) represents patients who eat three regular meals with more morning and afternoon snacks than the E1 patients. The E2 patients also have higher average per-meal carb intake than those in E1. The E3 cluster shown in Figure 4.4(c) represents patients who tend to eat throughout the day with no prominent frequency peaks and have lower carb intake per-meal when compared to the E1 and E2 patients. The number of patient-segments (out of all 92) that fall into the three Eat clusters are 28, 30, and 34, respectively.

"Trust" Behavior Analysis

The BWZ feature recommends a bolus dose each time the user activates it. In the CSII dataset, we iterate through the records of user-selected insulin boluses and compare the BWZ-recommended doses with the corresponding user selected doses. For each patient-segment, we aggregate all pairs of [BWZ-recommended dose, user-selected dose] and calculate the probabilities of the patient following, increasing, or decreasing the BWZ-recommended doses, as well as the magnitudes of dose adjustments. We then feed all patient-segments' BWZ-adherence profiles, each of which consists of the three probabilities, into a clustering algorithm and identify four clusters, each of which represents a group of patient-segments with similar BWZ-adherence patterns.

Figure 4.5 shows the aggregated box plots of the differences between the BWZ-recommended dose and corresponding user-selected doses in each Trust cluster. The clusters are denoted as clusters T1, T2, T3, and T4. The T1 cluster represents

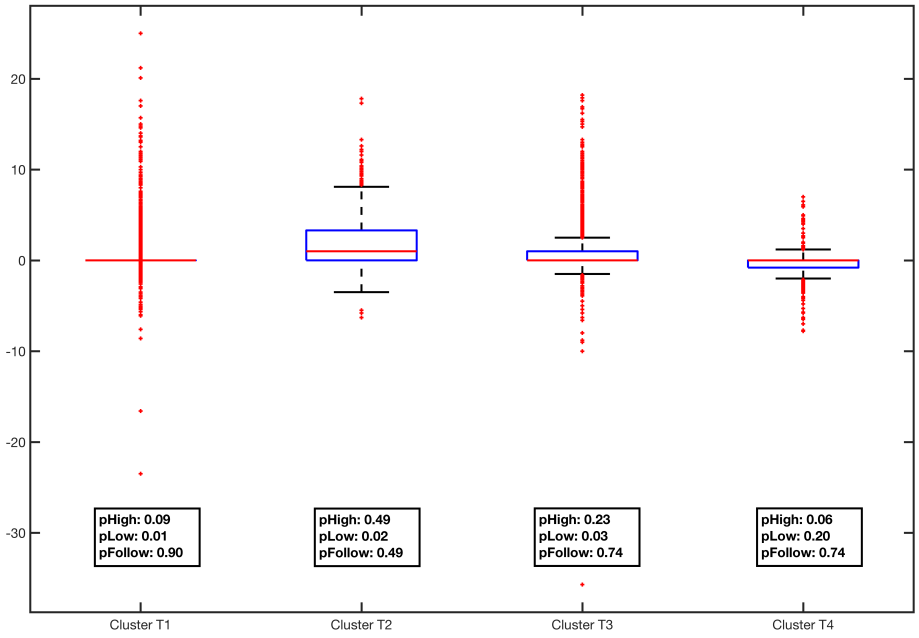


Figure 4.5: Box plots of the differences between user-selected and BWZ-recommended boluses of all patient-segments in four Trust clusters. The three probabilities shown for each cluster are the aggregated probabilities of patients increasing (pHigh), decreasing (pLow), or following (pFollow) the BWZ-recommended boluses. T1 shows a high probability of patients following the BWZ-recommended doses. T2 shows a high probability of patients increasing the BWZ-recommended doses. T3 shows a moderate probability of patients increasing BWZ-recommended doses. T4 shows a moderate probability of patients decreasing BWZ-recommended doses.

patients who strongly prefer following BWZ-recommended doses and only occasionally make adjustments. Note that in the box plot, samples are heavily condensed around the means because in most occasions the differences are close to zero. The T2 cluster represents patients who most of the times prefer higher doses than the BWZ-recommended values with significant dose increases. The T3 cluster represents patients who mostly follow the BWZ-recommended doses and sometimes make moderate adjustments, most of which are increasing the BWZ-recommended doses. The T4 cluster represents patients who mostly follow the BWZ-recommended doses

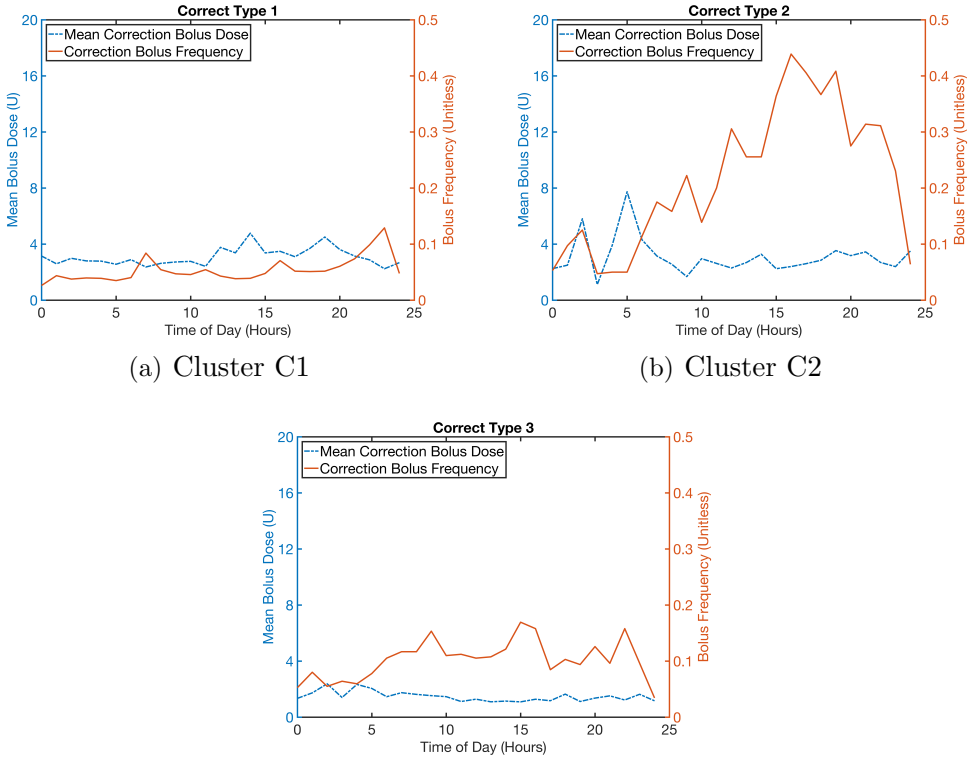
and sometimes make moderate adjustments, most of which are decreasing the BWZ-recommended doses. The number of patient-segments (out of all 92) that fall into the four Trust clusters are 53, 6, 25, and 8, respectively, suggesting that most patients either mostly strictly follow the BWZ-recommendations or make moderate incremental adjustments.

The T3 and T4 represent patients who tend to adjust BWZ-recommended doses in opposite directions. The clustering algorithm does not find a "negative image" to T2, which would represent patients who frequently make aggressive decreasing adjustments to the BWZ-recommended doses. This indicates that the BWZ-recommended dose calculation is tuned to be conservative for most diabetic patients, which makes sense from a safety standpoint: Insulin overdose can cause life-threatening hypoglycemia [15], and therefore, the BWZ software must be tuned to be safe for most patients. This also explains why few patients are in the T2 or T4 clusters: Given that BWZ is tuned to be conservative, most patients should not need to frequently decrease the doses (T4), and on the other hand, most patients should not need to over aggressively increase the dose either (T2).

"Correct" Behavior Analysis

Unlike mealtime boluses, which are associated with a patient's routine daily meal patterns, both the frequency and the doses of correction boluses highly depend on personal preferences and the patient's willingness as well as availability to manage blood glucose. From the CSII dataset, we calculate the aggregated distribution of correction bolus frequencies and doses over the whole hour times of a day (similar to how we treat the meal information, we round the bolus times to the nearest whole hour times). We feed the correction bolus frequencies and dose distributions into a clustering algorithm and identify three clusters of representative correction bolus patterns, denoted as clusters C1, C2, and C3.

Figure 4.6 shows the mean dose and frequency distributions of all the patients in



(a) Cluster C1

(b) Cluster C2

(c) Cluster C3

Figure 4.6: Correction bolus mean dose and frequency distributions of all patients in the four Correct clusters. C1 shows rare correction bolus use. C2 shows frequent correction bolus use with moderate doses. C3 shows occasional correction bolus use with three frequency peaks in a day.

each correction bolus cluster. Similar to Figure 4.4, the frequencies (red solid lines in Figure 4.6) represent aggregated daily frequencies: For example, 0.1 at 7 AM. in Figure 4.6(a) means for all patient-segments within cluster C1, on average 10% of the days a patient would take a correction bolus around 7 AM. The dashed dot blue line in the same figure represents the corresponding mean dose distribution over all the patients in the cluster.

The C1 cluster shown in Figure 4.6(a) represents patients who rarely take correction boluses. The C2 cluster shown in Figure 4.6(b) represents patients who frequently take correction boluses during the daytime with moderate doses. There are two notable dose peaks in the midnight to early morning period, indicating that

these patients sometimes take a large bolus during that time interval. The C3 cluster shown in Figure 4.6(c) represents patients who occasionally take correction boluses with three peak frequency times around 9 AM, 4 PM, and 10 PM. The number of patient-segments (out of all 92) that fall into the three Correct clusters are 55, 17, and 20, respectively, suggesting that most patients do not frequently take correction boluses. In all three clusters, the bolus doses are mostly in the low-mid range (0-5U). This is consistent with common clinical guidelines of diabetes self-management: Large boluses at non-mealtimes are usually not recommended as they may cause life-threatening hypoglycemia.

Summary and Remarks of the ETC Behavior Analysis

The CSII dataset includes T1D patients who visit the clinic during the data collection period starting in May 2014. We include a patient's data as long as the time ranges of the insulin data and CGM data at least partially overlap, because we need time-matched insulin and glucose data to individualize the physiological model for closed-loop evaluation, which is presented later in Section 4.3.5. We do not have any other patient screening criteria for data inclusion. The current CSII dataset includes T1D patients whose ages range from 23 to 79 and body weights range from 50 kg to 175 kg. As noted before, the set of patients represents the majority of T1D patients at the study site who use both insulin pumps and CGM sensors.

The three Eat clusters, four Trust clusters, and three Correct clusters generate 36 possible ETC combinatorial types. Table 4.1 lists the frequencies of the ETC types observed in the CSII dataset (the remaining ETC types not presented in the table are never observed on any of the patient-segment in the CSII dataset). The most frequent Trust and Correct combination is T1C1 (42% of patient-segments are this subtype), indicating that a significant portion of patients rarely make adjustments to the BWZ recommended doses and rarely take correction boluses. The T3 subtype is less common than T1 but still represents 27% of patient-segments. The T2 and T4

Table 4.1: Frequencies of ETC types in the CSII dataset.

ETC Type	Frequencies (of 92 patient-segments)
E1T1C1	17
E3T1C1	11
E2T1C1	11
E3T3C3	6
E2T3C1	6
E2T1C2	4
E1T3C1	4
E3T4C1	3
E3T1C3	3
E3T1C2	3
E2T3C2	3
E3T4C3	2
E3T4C2	2
E3T3C1	2
E2T2C2	2
E1T3C3	2
E1T1C3	2
E3T2C3	1
E3T2C2	1
E2T4C3	1
E2T3C3	1
E2T2C1	1
E2T1C3	1
E1T3C2	1
E1T2C3	1
E1T1C2	1

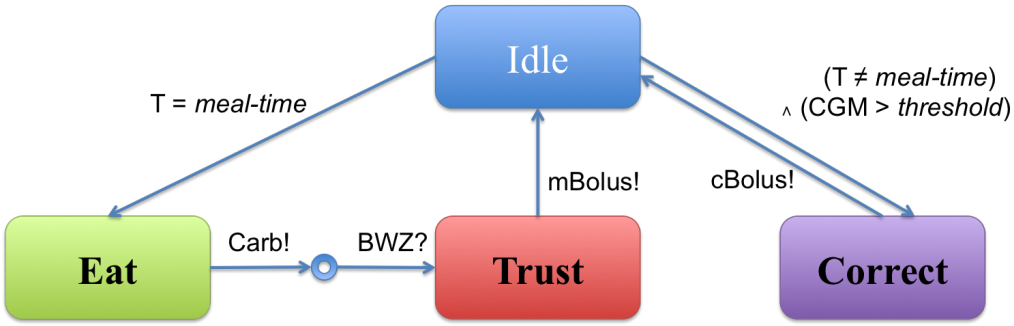


Figure 4.7: An overview of the user behavior.

subtypes represent uncommon subtypes, but they represent very distinct patterns than T1 and T3, and a number of patients do exhibit those Trust patterns. C2 and C3 are less common subtypes but do cover 41% of the patient-segments. Each of the three Eat subtypes covers about one-third of the patient-segments, indicating diverse and almost evenly distributed eating habits within the patient population.

The ETC model captures the most important user activities that directly impact the glucose control outcomes. In early 2015, we present the ETC data analytics results to a diabetes research group led by Dr. Michael Rickels, who is the director of Translational Research Program in Institute for Diabetes, Obesity & Metabolism at the University of Pennsylvania School of Medicine. The clinicians think that the ETC data mining approach extracts new information that is highly clinically relevant but is not part of the current vendor pump data analytics software outputs, which only include overall statistics such as the means and variances of CGM values. The clinicians believe that the ETC metrics provide important insights for caregivers to better understand how each patient's personalized glucose-related behavior impacts his/her glucose levels, which would ultimately promote the efficacy of treatment and improve clinical outcomes.

4.3.4 Data-Driven Behavior Modeling

The goal of modeling personalized behaviors is two-folds: (1) Discern clinically relevant insights about user behaviors from the vast amount of raw time-series data and represent the findings in a concise form such that the caregivers can easily apprehend; (2) Analyze how changing behaviors may impact the physiology and provide actionable feedback to the users. Achieving these goals requires developing a quantitative behavior model that is clinically meaningful and at the same time suitable for closed-loop analysis to provide actionable feedback to human caregivers and users.

This is a difficult research problem with several major challenges. First, constrained by practical limitations, the raw time-series behavioral information obtained from MCPS is typically noisy, sporadic, and limited in sampling duration & frequency. It was only in recent years that systems such as sensor-enhanced wearable pumps grow more popular, and the downloadable user activity logs become available. There currently lacks established techniques of representing and extracting behavioral features from this new type of clinical data. Second, the data-driven models must enable intuitive clinical interpretations at a certain level, such that the caregivers and patients can get actionable feedback from the model-based analysis. Third, there is a notable representation gap between the behavior information, which is currently in the raw sporadic time-series form, and the input format that most closed-loop analysis techniques require, which is well-defined math functions.

In this section, we present a methodology to address the afore-mentioned challenges and apply it to modeling the ETC user behaviors in the case study. We propose a technique to extract personalized probabilistic behavior features from the raw time-series data and analyze common patterns of the behavior features using a clustering technique. The behavior model representations have intuitive clinical interpretations. In addition, the behavior models can be integrated with physiological models in the closed-loop analysis that offers user feedback on how they may change behaviors to improve clinical outcomes.

Figure 4.7 illustrates the operational workflow in the use of the CSII system. At non-mealtime, the patient interacts with the system by requesting correction boluses. At mealtimes, the patient inputs the carb count and take meal-time boluses with the assistance of the BWZ feature. By applying the TAP-LT framework, we identify three main user behavior factors, "Eat, Trust and Correct". In the rest of this section, we describe our approach of building the ETC model from raw data and using it in closed-loop verification.

Quantitative Modeling of ETC Behaviors

In this section, we describe a personalized probabilistic representation to model each of the three behavior factors.

The meal carbohydrate intake and correction bolus usage both depend on the time of the day. To model the Eat behavior, we partition the time of the day into N_E intervals $T_1^E \dots T_{N_E}^E$, where $\cup_{i=1}^{N_E} T_i^E = [0, 24]$ (hours). We partition the possible value range of a carb count into M_E intervals $S_1^E \dots S_{M_E}^E$, where $\cup_{i=1}^{M_E} S_i^E = \mathbb{R}^+$. Similarly, to model the Correct behavior, we partition the time of the day into N_C intervals and partition the possible value range of a bolus dose into M_C intervals.

Let X_E denote the "Eat" matrix of dimension N_E by M_E and let X_C denote the "Correct" matrix of dimension N_C by M_C. An element x_{ij} in X_E represents the probability of the patient eating a meal with the carb count in the interval S_j^E within the time interval T_i^E. Similarly, an element x_{ij} in X_C represents the probability of the patient taking a correction bolus with dose in the interval S_j^C within the time interval T_i^C. Clearly, for each row, we have $\forall i \sum_j x_{ij} = 1$. We estimate the probability matrices X_E and X_C for each patient-segment from the CSII dataset.

The Trust behavior indicates a patient's BWZ-adherence level. We aggregate all the [BWZ recommended dose, user selected dose] pairs of each patient and estimate the probabilities of the patient increasing, following, or decreasing the BWZ-recommended doses: Those three probabilities are denoted as P_H, P_F and P_L, re-

spectively. Let D_T denote the probability distribution $D_T = \langle P_L, P_F, P_H \rangle$.

The complete ETC behavior model is a tuple $ETC_k = \langle X_E, X_C, D_T \rangle$, where k is the index of the patient data segment from which the model parameters are estimated. The ETC model is a compact quantitative representation that extracts the critical behavior metrics from the raw time-series data. The model parameters (i.e., the probability values) have clear practical meanings that facilitate clinical diagnosis and patient education. In current practice, when a Type 1 diabetic patient comes to the clinic for a routine visit, it is impossible for clinicians to manually go through a downloaded insulin pump log, which typical contains thousands of data points, and identify the trends in the raw data[14]. As a result, clinicians currently rely on a limited number of gross statistics such as overall mean and deviations of glucose readings to diagnose and make recommendations. Those statistics do not reveal temporal trend information of user behaviors, which the clinicians are interested in but lack methods and tools to extract from the data. The ETC model provides a means of analyzing and summarizing the key behavioral information in the raw time-series data.

Clustering of ETC Behavior Models

The previous section describes the ETC model that represents personalized probabilistic behavior traits. In this section, we aim at uncovering population-level common behavioral patterns from the patient-level ETC models. More specifically, we propose a learning technique that identifies clusters of patients who share similar behavioral traits. Population-level behavior classification provides practitioners with deep insights into the different behavioral types within a population and their distinct impacts on the clinical outcomes. Classifying patient behaviors also benefits patient education and diabetic community peer support, which provides vital op-

[14]As an example, in our study, an insulin pump log that is downloaded by clinicians during a routine visit typically contains several weeks of data. The CGM is sampled every five minutes, which translates into more than 2000 data points per week.

portunities for diabetes patients to share their own experiences in optimizing daily glucose-related behaviors to achieve better glycemic control.

To goal of clustering is to identify common patterns within the set of ETC_k tuples estimated on the CSII dataset. Note that ETC_k contains $N_E \times (M_E - 1) + N_C \times (M_C - 1) + 2$ free dimensions, which can quickly out-grow the size of the dataset (92 data segments in our study). Clustering high-dimensional data introduces a number of computational and theoretical challenges, as noted in extensive machine learning research [152], for example, the dimensions become hard to think of and visualize, computational intractability (the well-known "curse of dimensionality" issue), and the relative distance between samples converges as dimensions grow. Numerous machine learning techniques exist on tackling this challenge [3]. In the CSII dataset, not only that the number of free variables can become far greater than the sample size, the clinical meanings of the variables are highly heterogeneous and the three elements in ETC_k are estimated from disjointed subsets of the dataset. Therefore, we decompose the clustering problem into three sub-problems, i.e., clustering each of X_E, X_C, and D_T independently. The ETC clusters would consist of combinations of the sub-clusters.

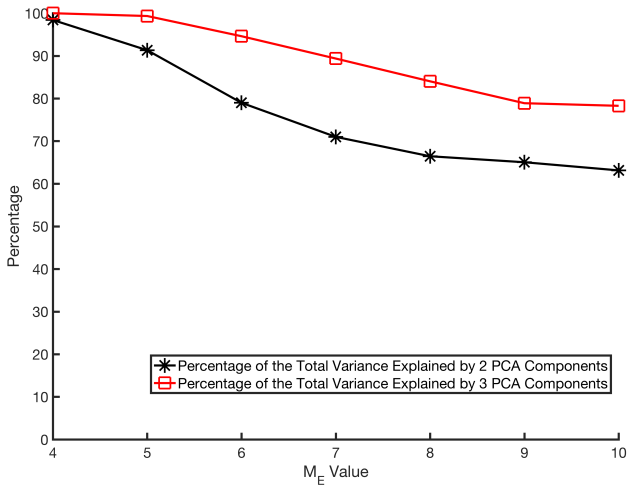
Dimension reduction of the Eat and Correct models. The "Eat" and "Correct" matrices X_E and X_C are each defined by two partitions: the time partitions and magnitude partitions. To tackle the dimensionality challenge, we design a two-step clustering technique. In the first step, we reduce the column dimensionality by transforming each row of X_E and X_C into a low-dimension representation that still retains most of the underlying information. In the second step, we conduct clustering analysis using the transformed compact representations of X_E and X_C.

The first step of our approach is based on a key observation about the characteristics of the X_E and X_C matrices. Each row in X_E or X_C is a distribution of input (either carb or insulin) magnitudes at a certain time interval. Although a

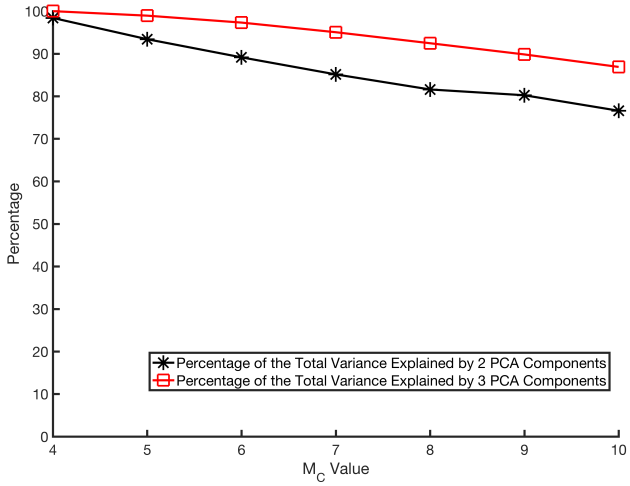
row contains M_E or M_C discrete probabilities ($M_E - 1$ or $M_C - 1$ are free variables as probabilities sum up to one), the fundamental feature of the whole underlying "distribution" may be characterized by fewer variables. For example, for a Gaussian distribution, regardless of the number of discrete partition intervals, the distribution can be characterized by two Gaussian parameters, i.e., the mean and deviation. This insight offers an opportunity to substantially reduce each row of the X_E and X_C matrices to a low-dimension representation which retains most of the information of the original row distribution.

To test whether the dimension-reduction insight can work on the estimated ETC model, we conduct principal component analysis (PCA) [132, 278] on the rows of each X_E and X_C matrices estimated from the 92 patient-segments given different configurations of M_E and M_C. Figure 4.8 shows the percentages of the total variances in the row data that are explained by the first two and three principal components in the PCA analysis, given different configurations of M_E and M_C. We can see that the first three principal components can explain more than 90% of total data variances in most M_C settings and can explain more than 80% of total data variances in most M_E settings. Intuitively, the PCA analysis results indicate that the information in each row distribution of the X_E and X_C matrices can be mostly captured using only two or three principal components after the PCA transformation.

We conduct PCA transformation on each row and reduce each 1-by-M_E or 1-by-M_C probability distribution to a 1-by-2 or 1-by-3 tuple that contains the transformed PCA coefficients of the first two or three principal components. As a result, the choices of M_E and M_C no longer affect the dimensionality of the transformed PCA representation, i.e., setting M_E and M_C differently will not increase the complexity of the subsequent clustering. Therefore, choosing M_E and M_C mostly involves trade-offs that stem from practical implications of the behavior model. On one hand, the partitions must distinguish meal carb counts and bolus doses at reasonable granularities so that caregivers can get clinically meaningful feedback from the model analysis,



(a) PCA Analysis of the Eat M_E Settings



(b) PCA Analysis of the Correct M_C Settings

Figure 4.8: PCA analysis results of different M_E and M_C settings. The figures show the percentages of the total variance in the rows in X_E or X_C that are explained by the first 2 and 3 principal components.

e.g., how patients may benefit from changing their eating and correction patterns. On the other hand, too fine-grained partitions cause data sparseness, which diminishes the power of PCA transformation. The intuition is that when the partition is overly fine-grained, the discrete probabilistic distribution becomes a sparse vector that no longer resembles a concentrated distribution shape, and thus the percentage

of variance explained by the first two or three principal components would decrease, as shown by the decreasing trend in Figure 4.8.

Partition configurations of the Eat and Correct models. Clinicians may set M_E and M_C within a reasonable range considering the context of the insulin pump application. For example, statistical analysis of the CSII dataset reveals that the carb intake per meal generally falls within 0 to 200 grams (most carb counts are under 100 grams). This is consistent with the general clinical recommendation that the daily total carb intake for an average adult should not exceed 300 grams, and significant reduction may be considered for Type 1 diabetics. The Eat matrix X_E divides the entire possible carb range into M_E intervals, and clinicians can choose M_E to achieve a particular level of granularity. For instance, $M_E = 4$ would characterize the carb intake with 4 intervals, which crudely distinguish low, medium, and high carb intake per meal. Increasing M_E enables more fine-grained distinction between different meal intake distributions. But note that the ETC model analysis would eventually be used to generate actionable feedback that is going to be implemented by humans, and therefore too fine-grained distributions not only causes data sparseness problem but are also not necessary or meaningful. Similarly, for the Correct matrix X_C, clinicians may set the number of intervals M_C considering the practical context: Most insulin bolus doses are within 0 to 15 Units, and patients mostly set the dose with a minimal increment (e.g., 0.5 Units). Overall, setting M_E and M_C between 4 and 10 appear to be a reasonable range considering the technical and practical trade-offs.

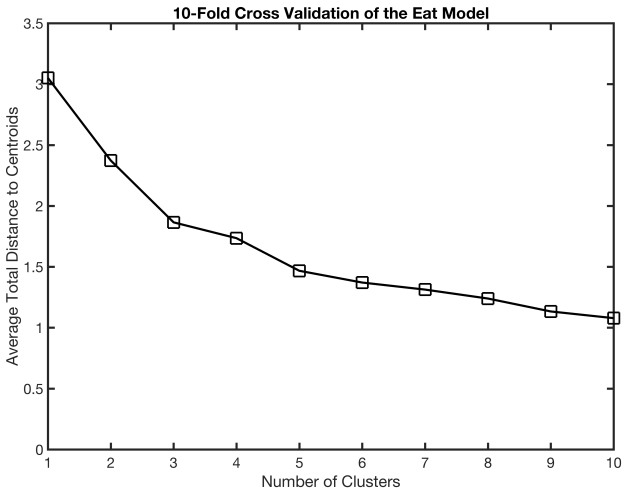
The same principles also apply to determining the number of time intervals in a day N_E and N_C. Since most eating and correction activities happen between 6 AM to 11 PM (excluding typical bedtime), there are a limited number of reasonable choices for determining how to partition the time of a day. For example, setting N_E and N_C to 3 means dividing the time intervals into morning, afternoon, and evening,

which offers a reasonable time-granularity for caregivers to interpret and use the ETC analysis results. On the other end, setting N_E and N_C to 6 would further distinguish six time intervals in a day, e.g., breakfast, morning snacks, lunch, afternoon snacks, dinner, and evening snacks. Dividing the time into even finer intervals would not only cause data sparseness problem but also make the ETC modeling results difficult to interpret.

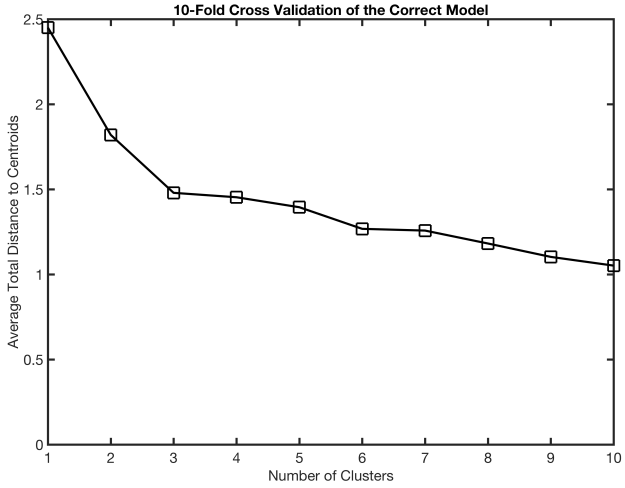
Clustering analysis of the Eat and Correct models. After the first step, each row in X_E or X_C matrix is transformed into a tuple $y_1, \ldots, y_n$ which represents the PCA coefficients of the first n principal components. We then stack the tuples of different rows by the time interval order and transform the X_E or X_C into a vector of length $n * N_E$ or $n * N_C$ (recall that N_E and N_C are the numbers of intervals, i.e., numbers of rows in the matrices). The transformed X_E and X_C, denoted as $X'_E = [y_{11}, \ldots, y_{1n}, \ldots, y_{N_E 1}, \ldots, y_{N_E n}]$ and $X'_C = [y_{11}, \ldots, y_{1n}, \ldots, y_{N_C 1}, \ldots, y_{N_C n}]$.

To cluster the Eat and Correct components, we run the k-means clustering algorithm [107] on the transformed X'_E and X'_C models. To determine the best number of clusters, we perform 10-fold cross validation [146] and measure the average total sum of squared distances to the centroids (i.e., the cost function) generated by k-means with different cluster settings. Figures 4.9(a) and 4.9(b) show how the cost function outputs over different numbers of Eat and Correct clusters. The curves exhibit a typical scree-plot pattern [252]: The cost value quickly decreases initially as the number of clusters increases, and the clustering algorithm fits the data better and better; then the cost value levels off past a certain turning point, indicating that over-fitting starts to happen. The turning point in the plot typically indicates a good number of cluster setting, as validated in other machine learning research [252]. For both Eat and Correct models, Figure 4.9 shows a clear turning point at 3 clusters.

To further validate the choices of number of Eat and Correct clusters, we conduct leave-one-out cross validation (LOOCV) [146, 140] with different numbers of clus-



(a) Cross Validation of the Eat Model



(b) Cross Validation of the Correct Model

Figure 4.9: Cross-validation results with different numbers of Eat and Correct clusters.

ters. By running LOOCV, we further test whether the clustering algorithm (given a particular number of clusters setting) is robust in the sense that leaving one sample out of the training set should not cause the clustering results to significantly vary. A cluster setting fails to pass the LOOCV if the clustering results fundamentally change over different runs, e.g., the centroids shift significantly, and/or a non-trivial

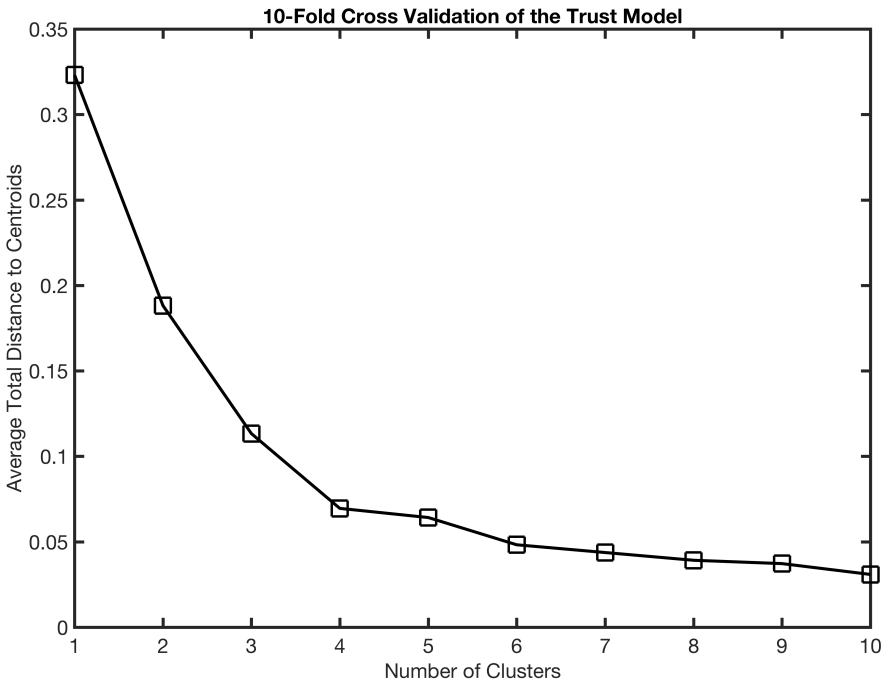


Figure 4.10: Cross-validation results with different numbers of Trust clusters.

portion of samples switch cluster identifications when a different single sample is being held out. The LOOCV results indicate that for both Eat and Correct models, the 3-cluster setting passes the LOOCV test, whereas 4 or more clusters would fail the LOOCV. This result is consistent with the 10-fold cross validation analysis: As the number of clusters increases past the turning point and starts to over-fit the data, some clusters will be increasingly more sensitive to holding one sample out. Therefore, we identify 3 Eat clusters and 3 Correct clusters. The statistics and clinical interpretations of the three Eat and Correct clusters are presented and discussed in Section 4.3.3 (see Figures 4.4 and 4.6).

Clustering analysis of the Trust model. To cluster the Trust component D_T, we run the k-means clustering algorithm [107] on the 92 "Trust" probability distributions D_T's. Similar to how we cluster the Eat and Correct models, we perform 10-fold cross validation to determine the number of Trust clusters, and we further

Table 4.2: Four output centroids returned by the k-means algorithm running over 92 patient-segments' Trust probability distributions D_T.

Centroid ID	P_H	P_F	P_L
1	0.09	0.90	0.01
2	0.49	0.49	0.02
3	0.23	0.74	0.03
4	0.06	0.74	0.20

Table 4.3: Statistics of the per-patient differences between the CSII glucose measurements and the model-simulated glucose values. BG denotes the blood glucose level.

Metric	Value
Mean Difference of Per-Patient Mean BG	15 mg/dL
Mean Difference of Per-Patient BG $>$ 180 Percentage	5%
Mean Difference of Per-Patient BG $<$ 70 Percentage	1%
Mean Difference of Per-Patient BG in [70,180] Percentage	6%

validate the result using LOOCV. Figure 4.10 shows how the cost function outputs over numbers of clusters, and it shows a clear turning point at 4 clusters. The LOOCV results show that the 4 cluster setting passes the LOOCV test, whereas 5 or more clusters would fail the LOOCV, which is consistent with the 10-fold cross validation analysis. Therefore, we identify 4 Trust clusters. Table 4.2 reports the four centroids. The four centroids correspond to the four Trust types T1 - T4 presented in Figure 4.5, respectively. The statistics and clinical interpretations of the four Trust clusters are presented and discussed in Section 4.3.3 (see Figures 4.5).

4.3.5 Probabilistic Verification of Behavior Models

An Individualized Physiological Model

We use a commonly accepted first-principle compartmental physiological model proposed in [27] and refer to it as the Bergman model. The Bergman model is a first-order differential equation that describes the interaction between the plasma insulin level and glucose level. To model the real-life scenario where a patient

Table 4.4: Comparison of population-wide glucose statistics of the CSII dataset and the model-simulated glucose data given the same insulin & meal inputs. BG denotes the blood glucose level. All BG outcomes are in the unit of mg/dL.

	CSII Dataset BG	Model Simulated BG
Mean of BG	161	155
Standard Deviation of BG	66	62
BG $>$ 180	32%	28%
BG $<$ 70	3%	3%

eats and takes subcutaneous insulin, the Bergman model can be augmented with a second-order meal pathway that relates ingested carbohydrates to plasma glucose appearance [99], and a second-order subcutaneous insulin pathway that relates subcutaneous insulin inputs to plasma insulin appearance [145, 212]. The details of the complete fifth-order augmented Bergman model are summarized in Chapter 5 Section 5.4.4. The model contains several physiological parameters that are patient specific: For example, the insulin sensitivity and basal glucose production rate. We tune the augmented Bergman model parameters within the value ranges reported in the original Bergman article [27], which include the glucose distribution rate, insulin sensitivity, meal glucose rate of appearance, and basal glucose production rate. For the other model parameters in the meal and insulin pathways, we use the nominal parameter values reported in the bio-medical research literature [212, 99].

For each patient in the CSII dataset, we feed the recorded insulin and carbohydrate inputs to the augmented Bergman model and compare the model-simulated glucose values with the CGM measurements in the CSII dataset. For each patient, we identify the model parameters on which the simulation best reproduces the key glucose outcome measures of the real data, which is a commonly accepted method of validating physiological models [182, 277]. Table 4.4 presents the key glucose statistics of the CSII measurement data and the simulated glucose outputs generated by the individualized Bergman model. The model is able to reproduce the key clinical

statistics, such as the mean and standard deviation of BG. We also compare the low and high glucose percentages, using the commonly accepted hypoglycemia threshold 70 mg/dL and hyperglycemia threshold 180 mg/dL [272]. Those are percentages of the glucose readings that are lower than 70 mg/dL and higher than 180 mg/dL, respectively. They are important clinical metrics of evaluating the risk of hypoglycemia and hyperglycemia, and are critical indicators of the quality of glycemic control. Table 4.4 shows that the individualized physiological model is able to reproduce the key glucose metrics in the CSII dataset. Table 4.3 presents the per-patient differences of the statistics of the model-simulated glucose values and the real glucose measurements in the CSII dataset. It shows that the model is able to reproduce the key glucose statistics not only at the population level but also at the individual level. We use the individually parameterized augmented Bergman model in the *in silico* closed-loop evaluation.

Closed-Loop Analysis

To demonstrate the utility of the proposed ETC modeling framework, we perform *in silico* experiments to examine the effects of patient behavior changes on the glycemic control outcomes. We model the closed-loop system by integrating the ETC behavior model, the individualized physiological model, and a model of the BWZ. We use the PRISM probabilistic model checker [157] to evaluate the hypoglycemia and hyperglycemia rates of different instantiations of the system model: Each instantiation is a pairing of a user behavior model and an individualized physiological model. PRISM is an open-source tool for formal modeling and analysis of systems that exhibit probabilistic behaviors. It can express and analyze several types of probabilistic models, such as discrete-time Markov chains, continuous-time Markov chains, and Markov decision processes. We encode the patient ETC behavior model as a discrete-time Markov chain, using probabilities derived from the CSII dataset. We use Matlab to generate finite traces from the simulation of the individualized augmented Bergman

Table 4.5: The effect of behavior (ETC types) change on the hypoglycemia and hyperglycemia rates for a patient with a high baseline hypoglycemia rate

	ETC Type	Hypoglycemia Rate (%)	Hyperglycemia Rate (%)
Actual type	E3T2C1	**6.93**	8.43
Change E subtype	E1T2C1	6.20	12.78
	E2T2C1	5.99	13.72
Change T subtype	E3T1C1	0.02	10.33
	E3T3C1	0.04	10.09
	E3T4C1	0.02	11.05
Change C subtype	E3T2C2	7.04	6.30
	E3T2C3	6.95	7.93
Change multi-subtypes	E2T1C1	0.04	16.46
	E2T2C1	5.99	13.72
	E3T1C3	0.10	9.76
	E2T1C3	0.08	15.42

model and encode them in PRISM. Then, we use PRISM to exhaustively check every possible execution trace of the closed-loop system and compute the expected hypoglycemia and hyperglycemia rates.

Given a patient's individualized physiological model, we pair it with his/her actual ETC type model or any other ETC types from the 36 possible combinations. We run the probabilistic model checking to evaluate the expected control outcomes of different pairings. The *in silico* experimental results (e.g., those reported in Table 4.5 and Table 4.6) identify the behavior changes that might help a particular patient improve the glucose control outcomes, i.e., reducing the hypoglycemia and/or hyperglycemia rate. The analysis results can benefit T1D patient education and diabetic peer support activities [110], in which groups of diabetic patients communicate and try to learn from each other's glycemic control experiences.

We evaluate the glucose control outcomes using the hypoglycemia and hyperglycemia rates. Here, we highlight the experimental results by showing the impact of ETC type changes on the hypoglycemia and hyperglycemia rates for two sample patients.

Table 4.6: The effect of behavior (ETC type) change on the hypoglycemia and hyperglycemia rates for a patient with a high baseline hyperglycemia rate

	ETC Type	Hypoglycemia Rate (%)	Hyperglycemia Rate (%)
Actual type	E1T1C1	0	**43.92**
Change E subtype	E2T1C1	0	44.38
	E3T1C1	0	41.62
Change T subtype	E1T2C1	0	39.13
	E1T3C1	0	43.46
	E1T4C1	0	45.31
Change C subtype	E1T1C2	0	41.59
	E1T1C3	0	43.47
Change multi-subtypes	E1T2C2	0	37.22
	E3T2C1	0	35.45
	E3T1C2	0	38.01
	E3T2C2	0	32.56

Table 4.5 presents the results for a patient with a high baseline hypoglycemia rate (6.93%) given his/her actual behavior type E3T2C1. As illustrated in Table 4.5, by only changing the E subtype, the hypoglycemia rate drops slightly. By changing the T subtype, the patient's hypoglycemia rate significantly decreases. The patient has a high likelihood of increasing the BWZ-recommended doses (the actual subtype is T2). The results in Table 4.5 suggest that if the patient follows the BWZ dose (T1) or even gives smaller doses (T4), the expected hypoglycemia rate would drop to around 0.02%, which would be a beneficial outcome. One fundamental challenge in glycemic control is that reducing the correction doses can mitigate hypoglycemia, but at the same time, it would also put the patient at a higher risk of hyperglycemia: For example, if the patient behaves as E3T1C1, then the expected hyperglycemia rate would be around 10.33%, which is slightly higher than the value of 8.43% with the actual type E3T2C1. Optimizing the insulin dose always comes down to balancing the risk of hypoglycemia and hyperglycemia. Hypoglycemia is a more critical short-term safety concern: Extreme hypoglycemia is life-threatening. Furthermore, the current population baseline hyperglycemia rate among Type 1 diabetics is in

the high range of 20% to 40% [272]. Therefore, significantly reducing hypoglycemia at the cost of slightly increasing hyperglycemia is justifiably beneficial to the patient. The results also show that changing the C subtype would not reduce the hypoglycemia rate for this patient. These experiment results could inform patient education: For example, clinicians may consider suggesting this patient to follow the BWZ-recommended doses more often, rather than frequently selecting higher doses.

Table 4.6 presents the results for another patient with a high baseline hyperglycemia rate (43.92%) in the experiments. The patient's actual behavior type is E1T1C1 and does not experience hypoglycemia. Based on the results, to reduce the hyperglycemia rate, the patient may consider reducing carbohydrate intake. For example, by changing the E subtype from E1 to E3, the expected hyperglycemia rate drops to around 41.62%. The treatment outcomes would be further improved if the patient increases BWZ-recommended doses (T2) or takes more correction boluses (C2), as highlighted in Figure 4.6. The optimal treatment outcomes can be achieved if the patient changes behavior in all three components of the ETC types: The expected hyperglycemia rate drops to 32.56% if the patient acts as type E3T2C2, which would be a significant improvement compared to the patient's baseline hyperglycemia rate.

The ETC model does not impose a major computational challenge on the probabilistic verification. Even given a fine-grained ETC model setting (N_E and N_C set to 6; M_E and M_C set to 8), the PRISM model checker running on an Intel(R) Xeon(R) CPU E5-2667 v2 @ 3.30GHz processor takes about 30 minutes to build the model (with about $120,000$ states and $200,000$ transitions) and then about 1 minute to perform the verification.

4.4 Summary of this Chapter

In this chapter, we designed a modeling methodology to analyze highly personalized user behaviors that are commonly observed in out-patient MCPS applications. We proposed the TAP-LT framework to systematic identify user behavior variables based on the analysis of a generalized architecture of patient-centered healthcare applications.

We applied the TAP-LT framework to an insulin pump case study and proposed an ETC probabilistic model that extracts the key behavioral trend information in the raw time-series clinical data. We developed a novel data-driven method to individualize the ETC model using clinical data and analyze common behavioral patterns at the population level. The proposed method includes a technique to reduce the model dimensionality for effective clustering analysis. We validated the ETC behavior model clusters by cross validation. The ETC model reveals novel quantifiable insights into the behavioral trends that can be used for personalized diagnosis. We demonstrated that the ETC model can be composed with an individualized physiological model in probabilistic verification, which enables in-silico analysis of how switching behavior patterns may improve clinical outcomes for certain patients. Such results can benefit patient education and patient peer-support.

Chapter 5

Validate Unreliable Behavior Information

In many human-in-the-loop systems, such as semi-autonomous driving [248] and user-supervised artificial pancreas [222], the automation agent monitors user behaviors at runtime so that it can adapt to critical behavioral events. The challenge is that the behavior measurements provided to the automation agent can sometimes be unreliable due to practical limitations, in which case incorrect behavior information may mislead the automation agent into taking improper actions that compromise safety. For example, some driver-assistance systems use computer vision techniques to infer whether the driver is distracted [248], which have inherent misdetection rates, and if the system incorrectly believes that a driver is distracted, it may engage emergency vehicle handling unnecessarily and cause "automation surprises" to the driver [122]. In artificial pancreas systems, if the user announces a meal to the bolus advisory feature and delays actual eating, it may trick the software into delivering insulin prematurely, which may cause life-threatening hypoglycemia. Therefore, it is critical to validate behavior information. The problem of validating behavior information in medical CPS is particularly challenging because physiological processes may contain non-linearities and patient-specific unobservable parameters (as

discussed before in Section 3.2.2), which implies that a behavioral event may trigger distinct physiological responses in different patients.

In this chapter, we propose a behavior event validation method that is designed to achieve a consistent detection performance despite parametric variances across individuals.

Part of the work described in this chapter has been presented in our previous paper [56].[15] Some of the new results may appear in our future publication [270].

The rest of this chapter is organized as follows: Section 5.1 formulates the behavior event validation problem; Section 5.2 introduces the parameter-invariant test in the medical system context; Section 5.3 presents a novel sequential decision filtering technique that exploits the temporal dynamics of the physiological process to achieve robust event detection; Section 5.4 applies the validation method to a meal detection case study. Evaluations using both an in-silico population and a clinical dataset validate that with the sequential decision filtering, the proposed detector outperforms three other existing meal detectors and achieves the lowest variances in all major performance metrics (without any individual tuning) despite inter-subject physiological variances; Section 5.5 concludes this chapter.

5.1 Problem Description

Using the control system notation convention, we represent the physiological process as a difference equation

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k), \mathbf{p}(k)), \mathbf{y}(k) = g(\mathbf{x}(k), \boldsymbol{\theta}(k)), \tag{5.1}$$

where $\mathbf{x}(k+1)$ is a vector of states at time step $k+1$, $\mathbf{u}(k)$ is a vector of inputs, $\mathbf{p}(k)$ is a vector of patient-specific model parameters, and $\mathbf{y}(k)$ represent the vector of measurements. Furthermore, let us assume $\mathbf{u}(k)$ consists of two parts $\mathbf{u}(k) =$

[15] Copyright retained by the authors.

$[u_\alpha(k), u_\beta(k)]$, where $\mathbf{u}_\alpha$ represents the inputs that the automation agent can reliably measure and $\mathbf{u}_\beta$ represents the unreliable inputs that need to be validated. For example, $\mathbf{u}_\alpha$ may be treatment activities that are recorded by medical devices, and $\mathbf{u}_\beta$ may be living activities that are patient self reported.

From the automation agent's perspective, at time step k, the information that is reliably available includes $\mathbf{y}(0) \dots \mathbf{y}(k)$ and $\mathbf{u}_\alpha(0) \dots \mathbf{u}_\alpha(k)$, where index 0 represents the start time. The automation agent needs to validate the input $\mathbf{u}_\beta(j)$ at some past time step j ($j < k$). The input $\mathbf{u}_\beta$ to be validated must be at a past time step j instead of current step k because the current input $\mathbf{u}_\beta(k)$ has not yet taken effects through the physiological process, i.e., the past measurements $\mathbf{y}(0) \dots \mathbf{y}(k)$ do not carry any information on the current input $\mathbf{u}_\beta(k)$. This is a fundamental property of "causal systems" [33], i.e., an output depends only on past inputs. Physiological systems, like many other physical processes, are causal in the standard control system formulation as presented here [191].

To simplify the presentation of the parameter-invariant test, without loss of generality, we assume the system model is formulated in a manner such that $\mathbf{u}_\beta$ is in a binary function form, i.e., $\forall k, \mathbf{u}_\beta(k) \in \{0, 1\}$. In some applications, the behavior validation questions are already binary decision problems such as "whether the driver is distracted" or "whether the patient started eating". We define the value to be validated is a "non-trivial" event represented by $\mathbf{u}_\beta(j) = 1$. Furthermore, we assume that if $\mathbf{u}_\beta(j) = 1$, then $\forall i \in w, \mathbf{u}_\beta(i) = 0$, where w represents a time interval that is adjacent to j and its size is a design parameter: We choose the window w to be smaller than the minimum separation between two non-trivial events (which depends on the application context) such that it is reasonable to assume that within the window w, at most one non-trivial event can happen, e.g., a patient can only start eating one meal within 30 minutes.

Using the notations introduced above, the behavior information validation problem is formulated as the following decision problem: Given a physiological pro-

cess in Equation 5.1 with possibly unknown, patient-specific parameters $\mathbf{p}$, the past measurements $\mathbf{y}(0) \dots \mathbf{y}(k)$ and past reliable inputs $\mathbf{u}_\alpha(0) \dots \mathbf{u}_\alpha(k)$, decide whether $\mathbf{u}_\beta(j) = 1$ at some past time step j.

5.2 Parameter-Invariant Test

In this section, we introduce the parameter invariant (PAIN) test, which is designed to achieve a constant false alarm rate (CFAR) despite inter-patient parameter variances. The fundamental idea of the PAIN test is to utilize a physiological model and trends in past measurements to capture the effects of unknown nuisance parameters, and then to establish invariance to the nuisance parameters by projecting the measurements onto a space which is unaffected by the unknown parameters, mathematically known as a null space projection. The benefit of the PAIN test is that the projected measurements will be the same, regardless of the patient's unknown parameters, allowing the design of powerful detectors that achieves population-level consistency. The PAIN test has been successfully applied to various engineering applications with unknown parameters [267, 269, 268] and has recently been extended to medical monitor design [56, 123, 236, 271].

Next, we formally describe the PAIN test. We start with a linear model of the physiological process. It is worth noting that although real physiological processes may contain some forms of non-linearities, linear modeling is still a very useful technique and has been successfully applied to many physiological modeling problems [62]. The fundamental trade-off is that non-linear systems, although can be more "realistic" in theory, are much more difficult to identify and analyze [242]. Following standard control theory techniques [213], a time-domain linear model can be transformed into a z-domain representation and then written in a discrete time matrix form $\mathbf{y} = \mathbf{F}\boldsymbol{\theta} + \mathbf{G}\boldsymbol{\upsilon} + \boldsymbol{\sigma}\mathbf{n}$, in which $\mathbf{y}$ represents the outputs, the matrix $\mathbf{F}$ represents the process model, $\boldsymbol{\theta}$ represents lumped physiological parameters and

parameters that correspond to reliable input $\mathbf{u}_\alpha$, $\mathbf{G}$ is a matrix that contains information on input $\mathbf{u}_\beta$, $\boldsymbol{\upsilon}$ represent inputs parameters that correspond to input $\mathbf{u}_\beta$, and $\boldsymbol{\sigma}\mathbf{n}$ is a zero-mean Gaussian noise.

The core of the PAIN test is a bi-directional hypothesis test. At each time step, to test the reported event $\mathbf{u}_\beta(j) = 1$, we divide the exclusive event window w (as described in Section 5.1) into two regions d_0 and d_1 such that $j \in d_0$. Our null hypothesis is that $\mathbf{u}_\beta(j) = 1$ and $\forall i \in w \wedge i \neq j, \mathbf{u}_\beta(i) = 0$, i.e., the non-trivial event indeed happens in the d_0 window. The event hypothesis is that $\mathbf{u}_\beta(j) = 0$ and $\mathbf{u}_\beta(j') = 1$, where $j' \in d_1$, i.e., the non-trivial event actually happens in d_1 instead of d_0.

Then we formulate the input matrix $\mathbf{G}$ according to the two hypotheses that map to a non-trivial event happening in time window d_0 or d_1. We write

$$\mathbf{H}_{k,i} = [\mathbf{F}_k, \mathbf{G}_i], i \in \{0, 1\}$$

and $\mathbf{H}_{k,i}$ spans the measurement subspace affected by the combined effect of parameters corresponding to the physiological dynamics, the reliable inputs $\mathbf{u}_\alpha$, and the hypothesized input $\mathbf{u}_\beta$ within the d_i time window. The physiological process model is then rewritten as $\mathbf{y}_k = \mathbf{H}_{k,i}\boldsymbol{\theta}' + \boldsymbol{\sigma}\mathbf{n}$, where $\boldsymbol{\theta}'$ contains the lumped physiological parameters $\boldsymbol{\theta}$ and input parameters $\boldsymbol{\upsilon}$. To be invariant to the unknown model parameters, we eliminate the effects of parameters $\boldsymbol{\theta}'$ by projecting $\mathbf{y}$ onto the null space of $\mathbf{H}_{k,i}$. Mathematically, the null space of an arbitrary matrix $\mathbf{X}$ is [117]

$$\langle \mathbf{X} \rangle^\perp = \{v | \mathbf{X}v = 0\}$$

and has an orthonormal basis transposed, $\mathbf{X}^\perp$, satisfying [117]

$$\mathbf{X}^\perp \in \{\mathbf{V} | \forall v \in \langle \mathbf{X} \rangle^\perp, \exists x, \mathbf{V}^\perp x = v \wedge \mathbf{V}\mathbf{V}^\perp = \mathbf{I}\}$$

where, $\mathbf{V}^{\perp}$ denotes the transpose of matrix $\mathbf{V}$ [117]. The following employs the above notation to present the test statistics. We introduce intermediate variables

$$\mathbf{r}_{k,0} = \mathbf{H}_{k,0}^{\perp}\mathbf{y}, \mathbf{U}_{k,0} = \mathbf{H}_{k,0}^{\perp}\mathbf{G}_1$$

$$\mathbf{r}_{k,1} = \mathbf{H}_{k,1}^{\perp}\mathbf{y}, \mathbf{U}_{k,1} = \mathbf{H}_{k,1}^{\perp}\mathbf{G}_0$$

where $\mathbf{r}_{k,0}$ and $\mathbf{U}_{k,0}$ denote the projection of the measurements and projected effect of $\mathbf{u}_\beta$ happening in d_1 onto the null space of $\mathbf{H}_{k,0}$, respectively (and vice-versa for $\mathbf{r}_{k,1}$ and $\mathbf{U}_{k,1}$). In words, $\mathbf{r}_{k,0}$ and $\mathbf{U}_{k,0}$ denote the measurements and the effects of the non-trivial event in d_1 which cannot be explained by physiological parameters $\boldsymbol{\theta}$, reliable inputs $\mathbf{u}_\alpha$, and the non-trivial event occurring within d_0. Consequently, to quantify whether the projected measurements and projected effects are significantly aligned [244], we write test statistics$t_i(\mathbf{y}_k)$ for $i \in \{0, 1\}$, as

$$t_i(\mathbf{y}_k) = \frac{r_{k,i}^{\perp}(\mathbf{I} - (\mathbf{U}_{k,i}^{\perp})^T(\mathbf{U}_{k,i}^{\perp}))r_{k,i}}{r_{k,i}^{\perp}(\mathbf{U}_{k,i}^{\perp})^T(\mathbf{U}_{k,i}^{\perp})r_{k,i}}$$

The form of $t_0(\mathbf{y}_k)$ is commonly referred to as an F-ratio in the signal processing and statistics literature [244], and has the useful feature that its value is invariant to the noise level of $\boldsymbol{\sigma}$ as well as the lumped physiological parameters $\boldsymbol{\theta}'$. In the context of this work, for $t_0(\mathbf{y}_k)$, the numerator denotes the magnitude of the projected measurements aligned with the projected non-trivial event effects of d_1 (i.e., in the space of $\mathbf{H}_{k,1}$), while the denominator represents the energy of the projected measurements which cannot be explained exclusively by non-trivial event effects of d_1, i.e., not in the space of $\mathbf{H}_{k,1}$. Thus, large/small values of $t_0(\mathbf{y}_k)$ implies that a non-trivial event within d_1 is likely/unlikely. Similarly, large/small values of $t_1(\mathbf{y}_k)$ implies that a non-trivial event within d_0 is likely/unlikely. Comparing $t_0(\mathbf{y}_k)$ to a threshold η_0, selected to achieve a specified probability of false alarm, generates a decision. Similarly, $t_1(\mathbf{y}_k)$ is generated by first projecting the measurements onto

Table 5.1: Score accumulation rules for $S(k)$.

	$t_0(\mathbf{y}_k) > \eta_0$	$t_0(\mathbf{y}_k) \leq \eta_0$
$t_1(\mathbf{y}_k) > \eta_1$	Event in d_0 or d_1 ($+t_1(\mathbf{y}_k)$ to each $j \in d_0$) ($+t_0(\mathbf{y}_k)$ to each $j \in d_1$)	Event in d_0 ($+2t_1(\mathbf{y}_k)$ to each $j \in d_0$)
$t_1(\mathbf{y}_k) \leq \eta_1$	Event in d_1 ($+2t_0(\mathbf{y}_k)$ to each $j \in d_1$)	No Meal (Do no change $S(j)$)

the null space of $\mathbf{H}_{k,1}$, then generating an F-statistic using $\mathbf{H}_{k,0}$ and comparing to a threshold η_1.

5.3 Sequential Decision Filtering

The aforementioned parameter invariant test generates a decision at each time step. At run-time, our detector runs in a sliding window fashion, with the relative positions of the windows fixed once the detector parameters are chosen. As the detector approaches a true event (the ground truth events are unknown to the detector), it will first pass through the d_0 window and then the d_1 window. Therefore, the event will accumulate a few d_0 decisions and then some d_1 decisions as the detector windows slide through. To leverage the structured sequential rise and fall of the statistics, we design an algorithm that generates a cumulative decision score based on the statistics $t_i(\mathbf{y}_k)$ for $i \in \{0, 1\}$. The statistics have the useful property that an increasingly positive $t_0(\mathbf{y}_k)$ implies a rising likelihood that an event has occurred in the window d_1 (and vice versa). Thus, the algorithm generates an S-score, $S(k)$, for each time step k (assuming k is initialized to zero) and accumulates S-scores according to the rules in Table 5.1, where a larger S-score indicates a higher confidence in the occurrence of an event.

At every step, when the detector claims an event occurs in window d_0, we add $2t_1(\mathbf{y}_k)$ to $S(j)$ for each time step j in the d_0 window; similarly, if the detector claims an event occurs in d_1, we add $2t_0(\mathbf{y}_k)$ to $S(j)$ for each time step in the d_1 window.

If it is likely that an event was in both windows, then we add $t_1(\mathbf{y}_k)$ to $S(j)$ for each time step in the d_0 window and similarly, we add $t_0(\mathbf{y}_k)$ to $S(j)$ of each time step in the d_1. Note that we drop the factor of 2 in the increments when both tests reject the null hypothesis, thus weakening the confidence of an event happening in any individual window. If both tests accept the null hypothesis, then neither d_0 nor d_1 is likely to contain a meal event; thus, no score accumulation occurs. In the end, peaks in the S-score curve indicate likely occurrences of non-trivial events. The thresholds (e.g., the height and width of a peak) that are used to define a positive event detection are tunable design parameters.

5.4 A Meal Detection Case Study

In this section, we apply the behavior information validation technique to the meal detection problem in diabetes care. This section is organized as follows: Section 5.4.1 motivates the problem; Section 5.4.2 summarizes the contributions of this case study; Section 5.4.3 states the problem formulation; Section 5.4.4 reviews glucose/insulin physiological modeling; Section 5.4.5 describes the PAIN meal detector design; Section 5.4.6 evaluates the meal detector and compares it to three existing detectors in evaluations using both an in-silico population and a clinical dataset.

5.4.1 Motivation

Type 1 diabetics depend on everyday insulin infusion or injection to maintain their glucose level within the acceptable range where too much insulin can cause life-threatening hypoglycemia, and too little insulin can cause nerve-damaging hyperglycemia [15]. Meal carbohydrates is a major disturbance factor to one's blood glucose level, and therefore every Type 1 diabetic faces a life-long control challenge: He/she has to carefully titrate insulin doses for every meal so that post-meal hyperglycemia is effectively controlled without risking hypoglycemia.

In recent years, Continuous Glucose Monitoring (CGM) technology has become more popular, which drives a whole new class of medical CPS, most notably the *artificial pancreas* (AP), that aims to facilitate glucose management for Type 1 diabetics. At the AP system's core are a CGM sensor, a wearable insulin pump for insulin infusion and boluses, and software that controls insulin titration [66]. Reliably predicting meals is difficult in real-life situations, thus all AP systems depend on certain kinds of meal declaration/detection mechanisms. Meal detection is a safety critical problem, where an incorrectly identified meal may trigger the system to either deliver too much insulin unnecessarily or deliver too little insulin, both of which have harmful (if not deadly) consequences.

Currently, Type 1 diabetics who use CGM sensors and wearable insulin pumps manually input the time and estimated carb count of each meal into the pump software, which then calculates a suggested insulin dose. Unfortunately, self-reported meal information is inherently unreliable [77]. Thus, more dependable meal detection methods are necessary to ensure patient safety. We propose a novel meal detection method that leverages a linear model of glucose and insulin responses that is inspired by a first-principle minimal physiological model [27].

5.4.2 Contributions

By applying the parameter-invariant test and sequential decision filtering technique, we develop a novel meal detection algorithm that is invariant to individual physiological parameters, i.e., it achieves a consistent detection performance across a patient population without needing individual tuning. We compare our meal detector with three other existing meal detectors [77, 109, 168] in evaluations using both an in-silico population and a clinical dataset. The in-silico and clinical evaluation results validate the unique strength of the PAIN detector: It consistently achieves the lowest variance (highest inter-subject consistency) in all major performance measures, including the false alarm rate, detection rate, and detection delay. In addition, the PAIN detector

achieves the shortest detection delay in the in-silico trial (detection delay cannot be analyzed in the clinical evaluation due to the lack of ground truth meal times) and better detection performance (measured by sensitivity vs. specificity) than the other three detectors across all operating points in the clinical evaluation.

5.4.3 Problem Statement

All AP systems need accurate estimates of the meal carbohydrate disturbances. To estimate the meal carbohydrate disturbances requires an accurate (and timely) estimate of when meals occur. Thus, the following summarizes our problem statement: We address the meal detection problem, where given recent glucose level measurements and insulin inputs, design a run-time monitor that accurately and quickly detects the onset of carbohydrates ingestion.

5.4.4 Glucose/Insulin Metabolism Models

First-principle models of glucose physiology broadly fall into two categories: maximal models and minimal models [63]. Maximal models use fine-grain compartmental submodels to describe the dynamics of glucose and insulin. These models are mostly used for simulation purposes, since controller design for non-linear models with unknown parameters is difficult. On the other hand, the minimal models use only a few coarse-grain compartments to model the physiology, and they have a relatively simple structure that is convenient for linearization and control design [99]. This section reviews existing glucose physiological models, including an FDA-accepted maximal model, which will later be used in *in-silico* evaluations, and a minimal model, which inspires the process modeling in designing our PAIN detector.

A Maximal Model

A maximal model that describes the glucose-insulin responses with meals has been proposed [187, 74], based on which the UVa/Padova Type 1 Diabetes Mellitus Metabolic Simulator (T1DMS) has been developed [151]. This model is an FDA-accepted substitute for animal testing in pre-clinical trials when evaluating certain control algorithms [75]. It consists of a set of continuous-time differential equations with 13 state variables and 32 physiological parameters. The model includes three sub-systems: the insulin subsystem, the meal glucose absorption, and the glucose kinetics. Due to space constraints, this section only sketches the model equations with brief explanations of the state variables. Table 3.6 lists the physiological meanings of some of the model parameters. Extensive details of the model, including the modeling rationale and meanings of the variables & parameters, can be found in a series of publications [187, 74, 151, 75].

The insulin sub-system describes the transportation of insulin from the subcutaneous injection site to other compartments of the body such as the liver, plasma, and tissues. This subsystem has seven state variables which evolve according to the follow equations [187, 74]:

$$\begin{aligned}\dot{I}_p(t) &= -(m_2+m_4)I_p(t) + m_1 I_l(t) + k_{a1}S_1(t) \\ &\quad + k_{a2}S_2(t)\end{aligned} \tag{5.2a}$$

$$\dot{X}(t) = P_{2U}/V_i I_p(t) - P_{2U}X(t) - P_{2U} * I_b \tag{5.2b}$$

$$\dot{I}_1(t) = k_i/V_i I_p(t) - k_i I_1(t) \tag{5.2c}$$

$$\dot{I}_d(t) = k_i I_1(t) - k_i I_d(t) \tag{5.2d}$$

$$\dot{I}_l(t) = m2 * I_p(t) - (m_1+m_3)I_l(t) \tag{5.2e}$$

$$\dot{S}_1(t) = -(k_{a1}+k_d)S_1(t) + u(t) \tag{5.2f}$$

$$\dot{S}_2(t) = k_d S_1(t) - k_{a2}S_2(t). \tag{5.2g}$$

In the above equations, I_p represents the mass of plasma insulin. $X(t)$ is a remote insulin signal that also appears in the glucose kinetics. I_1 and I_d represent a delayed insulin signal that governs the endogenous glucose production. I_l represents the liver insulin. S_1 and S_2 represent a two-compartment subcutaneous insulin process. $u(t)$ is the subcutaneous insulin injection/infusion input (wearable insulin pumps for Type 1 diabetics inject insulin into the subcutaneous tissue). Details about the parameters can be found in the maximal modeling literature [187, 74, 151].

The meal absorption sub-system models how meal carbohydrates pass the stomach, intestine and finally becomes glucose appearing in the plasma [74]. The stomach is represented by two compartments: one for the solid phase and the other for the liquid phase. The dynamics are modeled by the following equations [74]:

$$Q_{sto}(t) = Q_{sto1}(t) + Q_{sto2}(t) \tag{5.3a}$$

$$\dot{Q}_{sto1}(t) = -k_{gri}Q_{sto1}(t) + m(t) \tag{5.3b}$$

$$\dot{Q}_{sto2}(t) = -k_{empt}(Q_{sto})Q_{sto2}(t) + k_{gri}Q_{sto1}(t) \tag{5.3c}$$

$$\dot{Q}_{int}(t) = -k_{abs}Q_{int}(t) + k_{empt}(Q_{sto})Q_{sto2}(t) \tag{5.3d}$$

$$\begin{aligned} k_{empt}(Q_{sto}) = {} & k_{min} + (k_{max} - k_{min})/2 \\ & \times \left(\begin{array}{l} \tanh(\alpha * (Q_{sto} - b * D)) \\ -\tanh(\beta * (Q_{sto} - d * D)) + 2 \end{array} \right). \end{aligned} \tag{5.3e}$$

$$\alpha = 5/(2 * D * (1 - b)), \quad \beta = 5/(2 * D * d) \tag{5.3f}$$

Q_{sto} is the amount of glucose in the stomach. Q_{sto1} and Q_{sto2} represent glucose in the solid phase and liquid phase, respectively. Q_{int} is the amount of glucose in the intestine. $k_{empt}(Q_{sto})$ is a non-linear function that represents the rate of carbohydrates emptying from the stomach. D is the total amount of ingested glucose in the last meal. $m(t)$ is the input of meal carbohydrates.

The insulin and absorbed meal glucose interact through the glucose kinetics,

which is modeled by three state variables. The equations are given as follows [74]:

$$R_a(t) = f * k_{abs} * Q_{int}/BW \tag{5.4a}$$

$$\begin{aligned}\dot{G}_p(t) = &- k_1 * G_p(t) + k_2 * G_t(t) \\ &+ \max\left(0, k_{p1} - k_{p2} * G_p - k_{p3} * I_d(t)\right) \\ &- F_{snc} - \max\left(0, k_{e1} * (G_p(t) - k_{e2})\right) + R_a(t)\end{aligned} \tag{5.4b}$$

$$\begin{aligned}\dot{G}_t(t) = &- \frac{(V_{m0} + V_{mx} * X(t)) * G_t(t)}{K_{m0} + G_t(t)} \\ &+ k_1 * G_p(t) - k_2 * G_t(t)\end{aligned} \tag{5.4c}$$

$$\dot{G}_m(t) = - k_{sc} * G_m(t) + k_{sc}/V_g * G_p(t). \tag{5.4d}$$

G_p represents the plasma glucose concentration. G_t represents glucose in the rapidly equilibrating tissue. G_m represents the subcutaneous glucose. Note that the insulin action on glucose is modeled by $X(t)$ and $I_d(t)$ appearing in the $\dot{G}_p(t)$ and $\dot{G}_t(t)$ equations, and the meal glucose rate of appearance $R_a(t)$ is calculated from the meal sub-system state Q_{int}.

The maximal model of the T1DMS consists of all the 13 differential equations presented above. The insulin sub-system is a linear model. The meal sub-system contains a non-linear parameter $k_{empt}(Q_{sto})$. The glucose kinetics sub-system has several non-linear terms, such as the max operators and state product $X(t)G_t(t)$. Most of the model parameters, as listed in Table 3.6, are not identifiable. Because of the non-linearity and unknown parameters, it is very difficult to directly use the maximal model in control design. Instead, the model is used primarily for simulation purposes. The FDA accepted 300 virtual subjects, each of which is a realization of model parameters sampled from a joint distribution. The parameter distribution was drawn from clinical data obtained from individuals who underwent a triple tracer meal protocol in lab experiments [74].

Minimal Models

Minimal models represent another class of first-principle glucose-insulin models. The basic idea is to lump together compartments to describe the dominating dynamics of the glucose-related physiology using a minimal number of compartments. One of the most commonly accepted minimal models is described in [27], and referred to as the Bergman model. The Bergman model uses a single lumped compartment to model insulin and another lumped compartment to model glucose in plasma. Insulin governs the changes of glucose levels either directly or through another remote compartment. Under this compartmentalization scheme, at most three state variables are needed to describe the glucose-insulin physiology: plasma glucose, plasma insulin, and insulin in the remote compartment. Seven minimal models are proposed in [27], from the simple insulin-independent models (models No. 1 to 3), to the more elaborated forms (models No. 4 to 7). Not all the models use all of the three states. The Bergman model No. 4, whose equations are given as follows, has a linear form and explicitly describes insulin-dependent glucose uptake,

$$\dot{G}(t) = p_1 G(t) + p_2 * I(t) + p_3, \tag{5.5a}$$

where $G(t)$ and $I(t)$ represent plasma glucose and insulin, respectively, and p_1, p_2, and p_3 are model parameters.

While the Bergman model describes the plasma glucose-insulin dynamics, a second order, two-compartment meal pathway model is presented in [99]:

$$\dot{g}(t) = -\frac{1}{t_G} g(t) + \frac{A_G}{t_G} D_G(t) \tag{5.6a}$$

$$\dot{m}(t) = -\frac{1}{t_G} m(t) + \frac{1}{t_G} g(t), \tag{5.6b}$$

where $g(t)$ represents glucose in the first compartment and $m(t)$ represents the plasma glucose appearance, which is an input to the Bergman model. $D_G(t)$ is

the meal carbohydrate input. A_G is the carbohydrate bioavailability and t_G is the time of maximum glucose rate of appearance.

Lastly, the insulin pathway from subcutaneous tissue to plasma can be modeled by the following second order process [212, 145].

$$\dot{x}(t) = -k_a x(t) + u(t-\tau) \tag{5.7a}$$

$$\dot{I}(t) = -k_e I(t) + \frac{k_a}{V_d} x(t), \tag{5.7b}$$

where $x(t)$ and $I(t)$ are insulin in the subcutaneous compartment and plasma, respectively, k_a and k_e are rate parameters, V_d is the insulin volume, and $u(t-\tau)$ represents the insulin input with a time delay τ.

Combining equations 5.5a, 5.6a, 5.6b, 5.7a and 5.7b results in a fifth-order linear model that describes the glucose-insulin kinetics given meal carbohydrate inputs and subcutaneous insulin inputs. This linear model will be used in the PAIN detector design.

5.4.5 Meal Event Detector

In this section, we introduce the PAIN meal detector design. The remainder of this section details the bi-directional parameter-invariant test and the sequential decision filtering technique, respectively.

Parameter-Invariant Test

The design of PAIN detectors utilizing parameter invariant statistics originates from statistical signal processing [244]. Our primary goal lies in designing a detector (or monitor) that has consistent detection performance on a large population of Type 1 diabetics given a wide range of meal and insulin inputs. Because many of the patient-specific physiological parameters (e.g., those in the maximal models) cannot be identified, the detector is designed to be invariant to model parameters. Also, meal

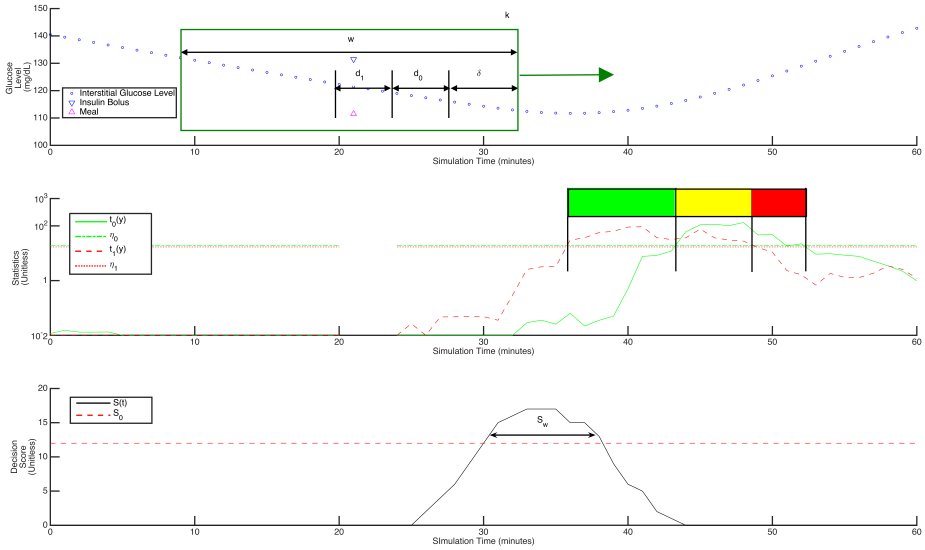


Figure 5.1: A meal detection example of the PAIN detector.

carb counts are manually reported by users in current Type 1 diabetes management systems, which can be unreliable for a variety of reasons, e.g., the patient may forget to report or miscount the meal portion. Therefore, the detector is also designed to be invariant to the exact magnitude of meal inputs. Insulin bolus times and doses are used by our detector since those are pump logged information.

We start with the linear minimal model in equations 5.5a, 5.6a, 5.6b, 5.7a and 5.7b. Following standard control theory techniques [213], the state-space linear model can be transformed into a z-domain representation and then written in a discrete time matrix form $\mathbf{y} = \mathbf{F}\boldsymbol{\theta} + \mathbf{G}\boldsymbol{\upsilon} + \boldsymbol{\sigma}\mathbf{n}$, in which $\mathbf{y}$ represents the outputs, $\mathbf{F}$ represents the process model, $\boldsymbol{\theta}$ represents model parameters, $\mathbf{G}$ represents input response, $\boldsymbol{\upsilon}$ represents inputs, and $\boldsymbol{\sigma}\mathbf{n}$ are zero-mean Gaussian noises.

The PAIN detector runs in a sliding window fashion. It has a few critical time windows which are illustrated in Figure 5.1. At each time step, the detector is given a vector of T past measurements $\mathbf{y} \in \mathbf{R}^T$ and insulin bolus inputs within the time window wt. To be invariant to the unknown model parameters, the PAIN detector

first eliminates the effects of parameters $\boldsymbol{\theta}$ by projecting $\mathbf{y}$ onto the null space of $\mathbf{F}$ so that the term $\mathbf{F}\boldsymbol{\theta}$ becomes zero.

The core of the detector is a bi-directional meal hypothesis test. The detector first hypothesizes that a meal happened wd steps back from the current time (meals are treated as impulses in the detector model). wd is a detector parameter that stays constant at run-time once it is chosen. The null hypothesis H_0 states that a meal indeed happened in a time window around next to the hypothesized meal time (the H_0 window in Figure 5.1). The event hypothesis H_1 states that a meal actually happened in an even earlier time window (the H_1 window in Figure 5.1). The sizes of H_0 and H_1 windows are design parameters to be chosen. The input response matrices G_0 and G_1 represent the hypothesized meal time windows of H_0 and H_1, respectively. When testing H_0 in the direction of H_1, the detector eliminates the effect of H_0 input by projecting G_1 onto the null space of G_0. It then calculates a statistic $t_0(y)$ as a ratio, where the numerator represents the remaining energy in the glucose measurements explained by signal under H_1 and the denominator represents the energy that is not explained by H_1. Then the detector tests H_1 in the direction of H_0 and calculates $t_1(y)$ in exactly the same way but in the opposite direction.

The statistic $t_0(y)$ assumes H_0 is true and tests H_1. The statistic $t_1(y)$ assumes H_1 is true and tests H_0. To guarantee a minimum level of performance, the detector rejects H_0 when $t_0(y) > \eta_0$, where η_0 is related the probability of false alarms. And similarly, the detector rejects H_1 when $t_1(y) > \eta_1$, where η_1 is related to the probability of missed detection.

The detector makes a meal detection decision at each time step based on the bi-directional hypothesis tests. When H_1 is rejected and H_0 is not, it claims a meal happened in the H_0 zone. When H_0 is rejected and H_1 is not, it claims a meal happened in the H_1 zone. When both are not rejected, it means there is not enough power in the signal to make a decision. When both are rejected, it means there are residual energy in the measurements that are explained by both hypotheses' signals.

Figure 5.1 demonstrates how the PAIN detector works on simulated scenarios generated by the FDA-accepted maximal model. The CGM measurements are sampled at one-minute time steps. The true meal happens around time 21 (the pink upper triangle in the figure). As the H_0 window approaches the true meal event (the detector never knows when a meal actually happened and runs hypothesis tests at every step), the statistic $t_1(y)$ (testing H_1 in the direction of H_0; the red dashed line in the figure) starts rising and becomes separated from $t_0(y)$. This indicates that H_1 is rejected and the detector claims H_0. Then as the detector moves further ahead, the true meal enters the H_1 window, and $t_0(y)$ starts rising and $t_1(y)$ starts falling, indicating that H_0 is rejected, and the detector claims H_1.

Sequential Decision Filtering

To apply the sequential decision filtering technique, we create a counting bin per each time step and register the S-score (as described in Section 5.3) under it, which represents the number of "meal hits". Table 5.1 presents the credit adding rules, which are also highlighted in Figure 5.1.

A peak in the S-score signal means that the detector makes a number of decisions at different time steps that all point to the same meal time, indicating a likely positive hit of a meal. On the other hand, if a bin only receives one or two counts (a typical width of H_0 and H_1 windows is 5 sample steps), it means the detector did not generate consistent decisions as the bin passed through the H_0 and H_1 windows, indicating a possibility of false positive. The sequential decision filtering technique mitigates such potential false positives, thereby improving the detection performance.

The S-score accumulation rules are highlighted in Figure 5.1: each colored region corresponds to the rule in Table 5.1 that applies in that region. In a typical positive meal detection scenario, one should first see the green region (corresponding to the d_0 window) approaches the meal event, followed by the yellow region as the meal

event transitions from d_0 to d_1, and finally see the red region, after which a peak in the $S(t)$ curve emerges, indicating that the detector makes a series of decisions at sequential time steps that all point to the same meal time region where the $S(t)$ peak emerges. The magnitude of $S(t)$ corresponds to our confidence in a meal occurring at time t. To trigger an alarm (indicating a meal has occurred), we utilize two design parameters, a threshold S_0 and a minimum width S_w; a peak is characterized by at least S_w consecutive $S(j)$s that are above S_0. At each time step, the detector raises a meal alarm if a new $S(t)$ peak emerges. The parameters S_0 and S_w can be tuned to achieve different detection performance: smaller S_0 and S_w result in higher sensitivity but more false alarms. We note that there is a few steps delay between the actual meal time and the $S(t)$ peak, as shown in Figure 1. This delay phenomenon is consistently observed in the in silico studies and is related to the physiological fact that there is a delay from the onset of eating to when the CGM reading starts changing: in the maximal model, meal carbohydrates have to pass several digestion compartments before affecting the plasma glucose.

5.4.6 Evaluation

We evaluate the PAIN meal detector using both an in-silico diabetes database and a clinical Type 1 diabetes dataset that is collected at the Hospital of the University of Pennsylvania. We compare the performance of our detector with three other existing meal detectors [77, 109, 168].

In-Silico Diabetic Database

We compare our PAIN detector with three existing meal detectors: the Dassau et al.s detector [77], Harvey et al.s detector [109], and Lee and Bequettes detector [168]. We evaluate the detectors in an in-silico clinical trial using the academic version of the FDA-accepted T1DMS simulator [151], which utilizes the maximal model. A "virtual subject" in the T1DMS simulator is a realization of the 32 physiological

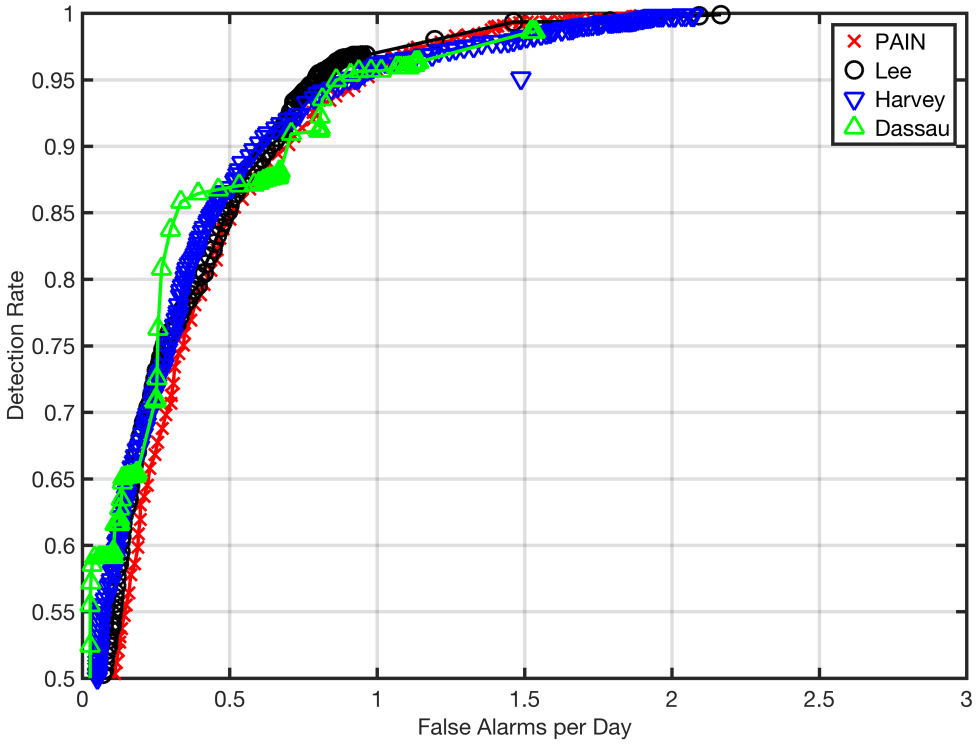


Figure 5.2: ROC curves of the four detectors in the in-silico evaluation.

parameters. The academic version of the T1DMS simulator contains 30 virtual subjects that are sampled from the same parameter distribution of the FDA-accepted population [75, 74].

The simulation configuration mimics the daily glucose management scenario of a T1D patient. Each virtual subject is fed three meals a day with randomized carb counts. The patient may check their glucose level at several checkpoints throughout the day and take correctional boluses if glucose levels are high. The bolus doses are calculated based on personalized diabetes profile parameters, e.g., insulin sensitivity parameters, which are included as part of the virtual subject specifications.

Table 5.2: Operating points of the four detectors in the in-silico evaluation.

False Alarms per Day	Detection Rates			
	PAIN	Lee	Harvey	Dassau
1.0	96%	97%	96%	96%
1.5	99%	99%	98%	98%

In-Silico Evaluation

We run the PAIN detector, Lee and Bequette's detector, Harvey et al.'s detector, and the Dassau et al.'s detector on the same continuous glucose measurements (CGM), which contain simulator-generated CGM noises, from the 30 virtual subjects in a 30-day in silico trial. Each of the four meal detectors has a set of configurable parameters, e.g., the decision score threshold of the PAIN detector and RoC thresholds of the RoC-based detectors. We systematically explore the combinations of each detectors parameters and get its best detection performance. A receiver operating characteristic (ROC) curve represents the detection rate and false alarm rate of a detector under different configurations.

Figure 5.2 shows the ROC curves of the four detectors. Table 5.2 lists two operating points of the four detectors. The operating points are chosen to compare the relative detection performance (sensitivity) of each detector for a chosen false alarm rate (specificity). In the figures and tables, the four detectors are abbreviated as "PAIN", "Lee", "Harvey", and "Dassau", respectively. The four detectors have very similar population-level detection performance in terms of the overall false alarm rates and detection rates.

The unique strength of the PAIN detector is that it is designed to achieve high inter-subject consistency regardless of physiological variances, whereas other RoC-based detectors may suffer from higher performance variances, especially on the outlier patients, due to the inherent limitation that threshold-based detection cannot account for real-time physiological variances. Figure 5.3 presents the inter-subject distributions of three key performance metrics in meal detection: The false alarm

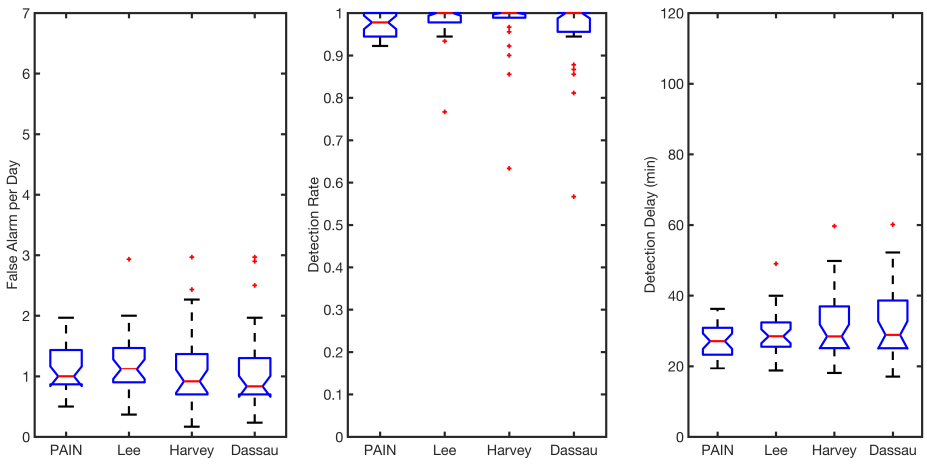


Figure 5.3: Box plots of the inter-subject performance distributions of the four detectors in the in-silico trial.

Table 5.3: Key performance metrics and their inter-subject variances of the four detectors in the in-silico evaluation.

Metric	Mean ± Standard Deviation			
	PAIN	Lee	Harvey	Dassau
False Alarm per Day	1.1 ± 0.4	1.2 ± 0.5	1.1 ± 0.65	1.1 ± 0.7
Detection Rate	$97\% \pm 3\%$	$98\% \pm 4\%$	$97\% \pm 7\%$	$96\% \pm 9\%$
Detection Delay (min)	$27 \pm 5\%$	$29 \pm 7\%$	32 ± 10	32 ± 11

rate, detection rate, and detection delay from the onset of a meal. The most notable feature in Figure 5.3 is that the PAIN detector has the lowest performance variance across all metrics with no outliers, whereas all other three detectors' suffer from significant performance degradation on certain outlier cases.

The in-silico evaluation validates the unique power of PAIN detection: The detector maintains a consistent performance over a physiologically heterogeneous population without any individual-level tuning. Table 5.3 lists the quantitative metrics that correspond to Figure 5.3. For a fair comparison, the operating points of all four detectors are set to achieve about 1 false alarm per day. The PAIN detector has the lowest variances of false alarm rate, detection rate, and detection delay. Also, the PAIN detector has the shortest detection delay among all four detectors.

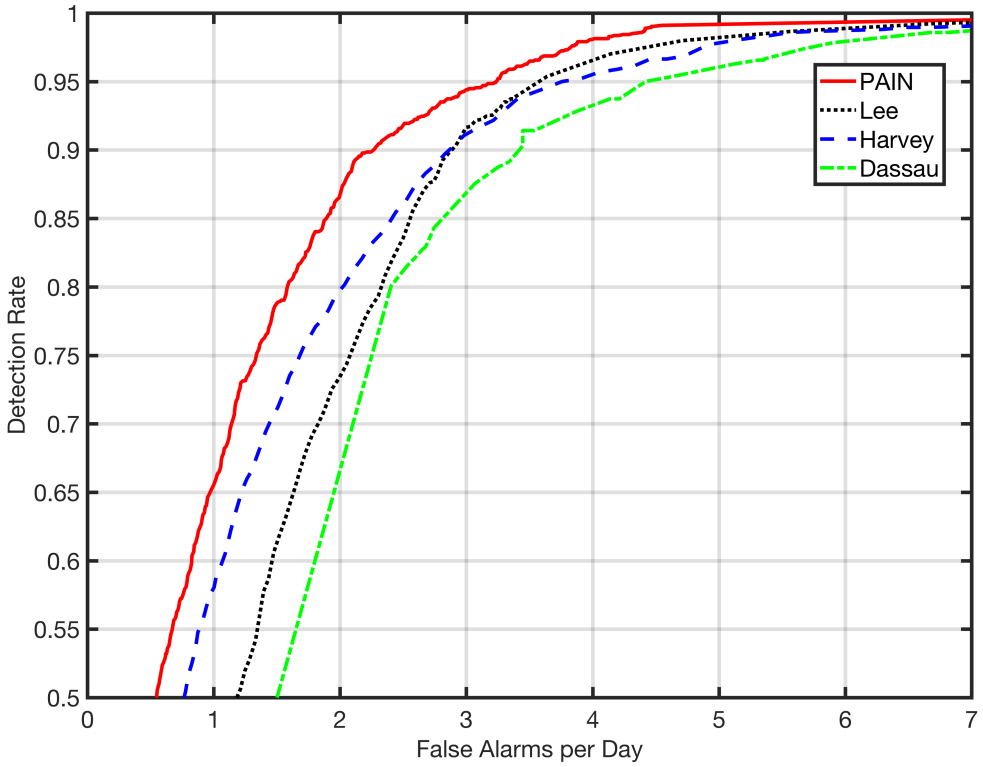


Figure 5.4: ROC curves of the four detectors in the evaluation using a clinical dataset.

Evaluation Using a Clinical Dataset

To further validate and compare the performance of the PAIN detector and the three other existing detectors, we evaluate them using a clinical dataset collected in the University of Pennsylvania Health System (with IRB approval). The clinical dataset includes five-minute CGM readings from 61 Type 1 Diabetes patients who use both CGM sensors and insulin pumps for daily diabetes management (mean ± standard deviation of age: 45.7 ± 15.3 years; mean ± standard deviation of body weight: 79.2 ± 21.9 kilograms; average duration of monitoring 17 days). The national registry of Type 1 diabetes patients receiving care in diabetes centers, of which Penn is a participating center, indicates that 60% of adults use insulin pumps, and 15% use CGM, and so from the 932 patients with Type 1 diabetes seen at the University of

Table 5.4: Operating points of the four detectors in the evaluation using a clinical dataset.

False Alarms per Day	Detection Rates			
	PAIN	Lee	Harvey	Dassau
2.5	92%	84%	86%	82%
3.0	95%	92%	91%	88%

Pennsylvania in the past year, 84 would be expected to use both an insulin pump and CGM. Thus, the 61 patients included represent the majority of patients expected to be utilizing sensor-enhanced pump technology in the management of their type 1 diabetes at the University of Pennsylvania.

Each patient counts the carbohydrates in the meal and then inputs that information into their insulin pump. The insulin pump will then provide a suggested meal bolus which the patient can accept or override. Since patient reporting of meal time (i.e., the time when the patient inputs the information into the pump) is error prone, we follow the commonly-accepted meal accounting rule [109] and consider any meal alarm within two hours of the reported meal event to be a correct detection (and a false alarm otherwise). In the event that a meal detector alarms within 30 minutes of a correction bolus (i.e., non-meal bolus), we omit this alarm from our false alarm (specificity) analysis since it corresponds to a "meal-like" critical event that the patient responded to by taking an insulin bolus, which indicates that an alarm could be useful.

We run the PAIN detector, Lee and Bequette's detector, Harvey et al.'s detector, and the Dassau et al.'s detector on the same glucose measurements from the 61 subjects. Figure 5.4 shows the ROC curves of the four detectors. Table 5.4 lists two operating points of the four detectors. The operating points are chosen to compare the relative detection performance (sensitivity) of each detector for a chosen false alarm rate (specificity). Figure 5.4 shows that for all false alarm rates (specificities), the PAIN detector has superior performance (higher sensitivities). Note that when a detection rate is in the high region (e.g., $> 85\%$), a seemingly moderate improvement

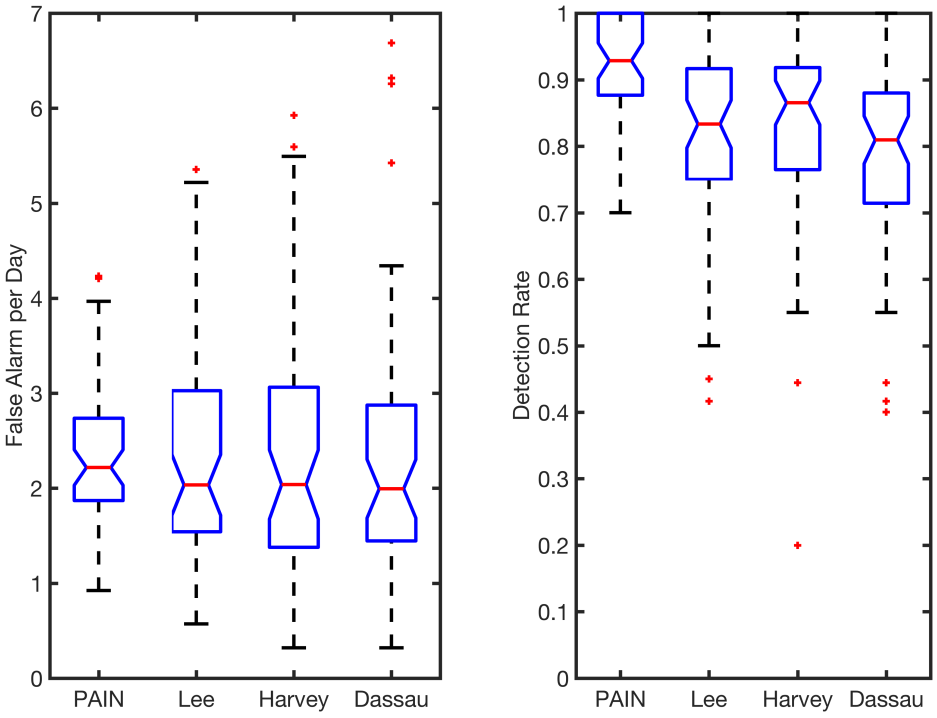


Figure 5.5: Box plots of the inter-subject performance distributions of the four detectors in the evaluation using a clinical dataset.

in detection rate may indicate a significant reduction in missed detections (e.g., a detector that achieves 95% detection reduces missed detections by 50% compared to another detector with 90% detection). We note that the evaluation results of the Harvey detector are comparable to those reported in the original publication [109].

Figure 5.5 compares the performance variability, in terms of false alarm and detection rates, of each meal detector on different patients in the dataset. We do not include detection delays in this comparison because, unlike in the in-silico trial, we do not have the ground truth meal times from the clinical dataset (patient-reported meal times are approximate and can be error-prone). These results provide a measure of the consistency of detection performance at the individual level, i.e., whether a detector can perform particularly bad on any subject. The duration of

Table 5.5: Key performance metrics and their inter-subject variances of the four detectors in the evaluation using a clinical dataset.

Metric	Mean ± Standard Deviation			
	PAIN	Lee	Harvey	Dassau
False Alarm per Day	2.4 ± 0.8	2.4 ± 1.2	2.4 ± 1.3	2.4 ± 1.5
Detection Rate	$91\% \pm 7\%$	$82\% \pm 14\%$	$84\% \pm 15\%$	$80\% \pm 15\%$

glucose monitoring varies across patients in the dataset. Eight patients are excluded from the individual-level analysis because their data contains fewer than 10 reported meals. Over the remaining 53 patients, the PAIN detector detects at least 70% of all reported meals and never has a false alarm rate greater than 4.4 false alarms per day. In sharp contrast to the PAIN detector, all other three detectors miss significantly more meals (both on average and worst-case), and have false alarm rates with higher variances and higher worst-case values. Table 5.5 lists the quantitative metrics that correspond to Figure 5.5. For a fair comparison, the operating points of all four detectors are set to achieve about 2.5 false alarms per day. The results show that the PAIN detector achieves the lowest variances in both false alarm rate and detection rate. In addition, the PAIN detector has the highest detection rate (sensitivity).

Discussion

The evaluation using the clinical dataset shows that the PAIN detector significantly improves the detection performance when compared with the other three detectors. For example, compared to the Harvey detector, the PAIN detector reduces the number of missed detections by 40% without increasing the false alarm rate in the evaluation. The population-level performance distribution results validate the unique strength of the PAIN detector: It is designed to be "invariant" to differences in patients physiological parameters and thereby achieves the lowest variance of all performance measures over a real patient population. This unique feature of the PAIN detector is critical to the glucose control applications: A meal detector that frequently misses true meal events on some subjects could result in severe post-

prandial hyperglycemia and possibly subsequent hypoglycemia "overshoots" of large insulin boluses (to correct the high glucose level). Due to the high inter-subject physiological variance of the diabetic population, one of the most critical technical challenges and a prominent regulatory concern is validating that a glucose control system is reasonably safe for every patient within the target population. With its unique feature of high inter-subject consistency, the PAIN detector can complement other research efforts that focus on achieving high control performance (but potentially with high inter-patient variances).

In theory, the performance of the RoC-based detectors may be further improved by carefully tuning the detector parameters for each individual patient. However, such tuning process may require frequent clinic visits and lab tests (which is not practically feasible for most patients) because patients physiological characteristics change over time. Moreover, even with parameter tuning, the RoC-based meal detectors have their fundamental limitation because the post-meal glucose rise rate depends on many other varying factors such as the nutrition composition of meals [279] and insulin-on-board [88], which can not be mitigated by simply tuning the threshold parameters. In contrast, the evaluation results demonstrate that the PAIN detector is able to achieve low-variance performance without any individual-level parameter tuning.

In terms of false alarm rates and detection rates (the ROC curve), we observe that the four detectors have similar ROC performance in the in-silico evaluation, however, in the evaluation using the real patients' data, the PAIN detector has superior performance to other three detectors with a notable margin across all operating points. Note that the clinical dataset that is used for evaluation represents how Type 1 diabetic patients manage glucose in daily use scenarios with various real-life disturbance factors that are not modeled in the T1DMS simulator, such as physical exercises, stress levels, other medications, and intra-patient physiological variances. All those factors can impact glucose physiology and cause glucose level fluctuations

that appear to be "meal-like" events from a detector's perspective. In the in-silico database, the glucose changes are simulated by a mathematical model in which insulin and meals are the only driving forces. Therefore, the compound physiological variances of the clinical dataset (considering all the real-life disturbances) can be greater than the simulated in-silico database. Additionally, the glucose readings in the clinical dataset are measured by real CGM devices, and the T1DMS simulator uses an auto-regressive CGM noise model, which may not represent all the sensor artifacts. This explains why all detectors exhibit a performance degradation when being evaluated on the real patient data. Comparing the in-silico and clinical evaluation results, the other three existing detectors take much greater performance hits than the PAIN detector. This further validates the unique advantage of the PAIN detector: It is more resilient to real-life physiological disturbances than other existing detectors.

5.5 Summary of this Chapter

In this chapter, we have proposed a methodology to validate safety-critical behavior events in MCPS that may be error prone for practical reasons, e.g., user report errors. The core of the method consists of an application of the parameter invariant (PAIN) detection theory enhanced by a novel sequential decision filtering technique. The unique advantage of the proposed method is that its detection performance is invariant to uncertain physiological parameters, thereby achieving a high performance consistency despite inter-patient physiological variances.

We applied the proposed validation method to a meal detection case study and designed a PAIN meal detector. We compared the PAIN detector with three other existing meal detectors in both an in-silico trial and evaluation using a clinical dataset. The evaluations validate the unique strength of the PAIN detector: Compared to the three existing detectors, the PAIN detector not only achieves the best detection

performance (measured by the false alarm rate, detection rate, and detection delay) but also has the lowest performance variance (highest consistency) in all performance measures. The evaluation results indicate that the PAIN method is a promising technology in achieving consistent detection performance despite significant inter- and intra-subject physiological variances, which is one of the most critical technical and regulatory challenges for the artificial pancreas (AP) research.

Chapter 6

Conclusion

6.1 Summary of this Dissertation

In this dissertation, we have developed methodologies for modeling different types of user behaviors in MCPS and using the behavior models for system analysis. Chapters 3 , 4, and 5 present three research thrusts, respectively:

- In the first research thrust (Chapter 3), we have designed a model-based analysis framework for evaluating generic (non-personalized) behaviors that are typically driven by rule-based protocols. By applying the method to an intra-operative glycemic control case study, we identified limitations of a current clinical protocol, designed an enhancement, and formally verified that the new protocol is robustly safe for a virtual population of an FDA-accepted physiological model that is instantiated to continuous ranges of uncertain physiological states and parameters. To cope with the practical challenge that a patients physiological parameters may exhibit transient fluctuations in real surgical scenarios, we have developed a run-time safety monitoring technique to adaptively track the real-time physiological responses using the maximal model and generate predictive alarms on critical events. Evaluation using a clinical dataset shows that the proposed prediction algorithm achieves a high sensitivity with

a low average false alarm rate.

- In the second research thrust (Chapter 4), we proposed a TAP-LT framework to systematically model personalized behaviors that are commonly observed in patient-centered healthcare applications. We applied the TAP-LT framework to an insulin pump case study and proposed an ETC probabilistic model that extracts personalized behavioral trend information from the clinical data. We developed a novel data-driven method to individualize the ETC model and a clustering technique to analyze population-level behavioral patterns in the presence of the model dimensionality challenge. We demonstrated that the personalized model not only reveals new clinically relevant behavioral trends but also enables closed-loop verification that provides quantitative insights into how certain users may achieve better physiological outcomes by switching behavioral pattern.

- In the third research thrust (Chapter 5), we have proposed a methodology to validate safety-critical information in MCPS that is potentially unreliable. The proposed method is designed to be invariant to uncertain variances in individual physiological parameters, thereby achieving high inter-subject consistency of detection performance. We applied the method to a meal detection case study and compared the novel PAIN meal detector with three major existing meal detectors. Both in-silico and clinical evaluations validate that the PAIN detector's performance has the lowest population-level variances and is superior to all existing detectors.

6.2 Future Research Opportunities

The emerging trend of "ubiquitous healthcare" is driving rapidly advancing innovations in the healthcare industry that aim at continuously monitoring a person's health status and providing real-time feedback to improve the quality of life in var-

ious types of living scenarios. Those applications interact with users at multiple physiological, behavioral, and psychological levels. The increasingly complex interaction between the user and technology presents numerous engineering challenges and research opportunities. This dissertation focuses on developing model-based techniques to represent, analyze, and validate different types of behavioral factors in MCPS. There are plenty of future research opportunities in each of the three research thrusts described in this dissertation and in achieving possible synergy between them. In the rest of this section, we discuss several research directions both within and across the three research thrusts.

Thrust 1: Model-Based Analysis of Generic Behaviors

In the case study presented in Section 3.3, after being initially evaluated by simulation, the new protocol successful passes the formal verification over the virtual population in dReach. In general, the protocol design and verification could be an iterative process in which counter examples identified in the verification may guide protocol revision. Future research may further explore how to systematically and automatically improve the control protocol design using counterexamples.

The formal verification results in Section 3.3.5 suggests that allowing all physiological parameters and states of the non-linear maximal model to simultaneously vary within the full over-approximated ranges presents a computationally challenging problem for a state-of-the-art hybrid system model checker. The proposed closed-loop system serves as a benchmark to motivate future research in improving hybrid system verification tools.

The maximal physiological model has been predominantly used for proof-of-concept, simulation-based evaluation of controllers [63]. The data-driven safety monitoring technique presented in Section 3.3.6 provides novel insights into the possibility of using the maximal model and its virtual subject set for run-time prediction and control. In this dissertation, we evaluated the technique using a retrospective clinical

dataset and showed that the algorithm achieves high sensitivity with a low average false alarm rate. A major future research opportunity lies in providing bounded performance guarantees that the sampled high-dimensional virtual subject set is sufficient to ensure safety.

Thrust 2: Model-Based Analysis of Personalized Behaviors

In Section 4.3, we have instantiated the TAP-LT framework as an ETC model in the case study. There are several interesting research directions in extending the scope of the behavior model. First, the ETC model can be expanded to represent time-varying behaviors. Analyzing and validating time-varying behaviors would require an extended clinical dataset that includes several long-term data segments on each patient. Second, the individualized ETC model information can be associated with other contextual information to generate more clinical insights: For example, whether a patient's Eat, Trust, and Correct behaviors are correlated with his/her past experience with the technology and other personal living habits. Third, the ETC model can serve as a virtual patient testbed for evaluating different artificial pancreas designs, especially those that supports user-adaptive multi-level automation [222].

Thrust 3: Validate Unreliable Behavior Information

In Chapter 5, we applied the proposed validation technique to the meal detection case study, in which we focused on detecting the presence of critical events. In many user-supervised medical applications, the first and foremost challenge is to detect critical events so that humans can take actions accordingly. As the healthcare applications start to adopt higher automation, a useful extension of the proposed method is to also estimate the quantitative magnitude of the events, e.g., meal carb amount, which would enable precise automatic control.

Potential Synergy Between Different Thrusts

The validation technique proposed in the third thrust has been validated in in-silico and retrospective clinical trials. An interesting research problem is to integrate the validation technique into a formal verification framework that is similar to the one described in the first thrust and establish bounded detection performance guarantees. In the meal detection case study context, this could mean formally verifying that the PAIN detector, which hypothesizes a linearized physiological model, can guarantee bounded detection performance over a virtual population of the non-linear maximal model. Such result can provide important novel insights into the role of minimal and maximal models in control design. Solving this problem would probably require substantial new theoretical research to transform the mathematical formulation of the validation algorithm into a form that can be ported into the verification tools.

Another synergistic research opportunity lies in combining aspects of all three thrusts. The behavior modeling technique proposed in the second thrust enables categorizing patients into different behavioral types. An interesting research problem is to utilize the patient's behavior information to fine-tune the data-driven safety monitoring technique described in the first thrust and the validation technique developed in the third thrust. Given the wide ranges of physiological and behavioral variances in the general patient population, it is very challenging (if possible at all) to develop a one-size-fits-all estimation, prediction, and control scheme. The individualized behavior modeling provides an opportunity to design safety controllers and validation techniques that are adaptive to different groups of patients.

Bibliography

[1] Gregory D Abowd, Hung-Ming Wang, and Andrew F Monk. A formal technique for automated dialogue development. In *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, pages 219–226. ACM, 1995.

[2] Mark Abramowicz, Gianna Zuccotti, and Jean-Marie Pflomm. Minimed 530g: An insulin pump with low-glucose suspend automation, 2015.

[3] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.

[4] Nisar Ahmed and Mark Campbell. Multimodal operator decision models. In *American Control Conference, 2008*, pages 4504–4509. IEEE, 2008.

[5] Nisar Ahmed, Ewart de Visser, Tyler Shaw, Amira Mohamed-Ameen, Mark Campbell, and Raja Parasuraman. Statistical modelling of networked human-automation performance using working memory capacity. *Ergonomics*, 57(3):295–318, 2014.

[6] Nisar Razzi Ahmed. *Probabilistic modeling and estimation with human inputs in semi-autonomous systems*. PhD thesis, Cornell University, 2012.

[7] Amos Albert. Comparison of event-triggered and time-triggered concepts with regard to distributed control systems. *Embedded World*, 2004:235–252, 2004.

[8] N A Ali, J M Jr O'Brien, K Dungan, G Phillips, C B Marsh, S Lemeshow, A F Jr Connors, and J C Preiser. Glucose variability and mortality in patients with sepsis. *Crit Care Med*, 36(8):2316–2321, 2008.

[9] Rajeev Alur, Robert K Brayton, Thomas A Henzinger, Shaz Qadeer, and Sriram K Rajamani. Partial-order reduction in symbolic state space exploration. In *Computer Aided Verification*, pages 340–351. Springer, 1997.

[10] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A Henzinger, P-H Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical computer science*, 138(1):3–34, 1995.

[11] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.

[12] A Anabtawi, M Hurst, M Titi, S Patel, C Palacio, and K Rajamani. Incidence of hypoglycemia with tight glycemic control protocols: a comparative study. *Diabetes Technol Ther*, 12(8):635–639, 2010.

[13] Y M Arabi, O C Dabbagh, H M Tamim, A A Al-Shimemeri, Z A Memish, S H Haddad, S J Syed, H R Giridhar, A H Rishu, M O Al-Daker, S H Kahoul, R J Britts, and M H Sakkijha. Intensive versus conventional insulin therapy: a randomized controlled trial in medical and surgical critically ill patients. *Crit Care Med*, 36(12):3190–3197, 2008.

[14] David Arney, Miroslav Pajic, Julian M Goldman, Insup Lee, Rahul Mangharam, and Oleg Sokolsky. Toward patient safety in closed-loop medical device systems. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pages 139–148. ACM, 2010.

[15] American Diabetes Association et al. Hypoglycemia (low blood glucose). *URL http://www.diabetes.org/living-with-diabetes/treatment-and-care/blood-glucose-control/hypoglycemia-low-blood.html*, 2011.

[16] J Elin Bahner, Anke-Dorothea Hüper, and Dietrich Manzey. Misuse of automated decision aids: Complacency, automation bias and the impact of training experience. *International Journal of Human-Computer Studies*, 66(9):688–699, 2008.

[17] Lisanne Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983.

[18] E H Baker, C H Janaway, B J Philips, A L Brennan, D L Baines, D M Wood, and P W Jones. Hyperglycaemia is associated with poor outcomes in patients admitted to hospital with acute exacerbations of chronic obstructive pulmonary disease. *Thorax*, 61(4):284–289, 2006.

[19] S Basnyat, P Palanque, R Bernhaupt, and E Poupart. Formal modellingof incidents and accidents as a means for enriching training material for satellite control operations. *Safety, Reliability and Risk Analysis: Theory, Methods and Applications (4 Volumes+ CD-ROM)*, page 45, 2014.

[20] Ellen J Bass, Matthew L Bolton, Karen Feigh, Dennis Griffith, Elsa Gunter, William Mansky, and John Rushby. Toward a multi-method approach to formalizing human-automation interaction and human-human communications. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[21] Ellen J Bass, Karen M Feigh, Elsa Gunter, and John Rushby. Formal modeling and analysis for interactive hybrid systems. In *Fourth International Workshop on Formal Methods for Interactive Systems: FMIS*, volume 45, 2011.

[22] Rita Basu, Barbara Di Camillo, Gianna Toffolo, Ananda Basu, Pankaj Shah, Adrian Vella, Robert Rizza, and Claudio Cobelli. Use of a novel triple-tracer

approach to assess postprandial glucose metabolism. *American Journal of Physiology-Endocrinology And Metabolism*, 284(1):E55–E69, 2003.

[23] Roy W Beck, William V Tamborlane, Richard M Bergenstal, Kellee M Miller, Stephanie N DuBose, and Callyn A Hall. The T1D Exchange clinic registry. *The Journal of Clinical Endocrinology & Metabolism*, 97(12):4383–4389, 2012.

[24] Beatrice Berard, Michel Bidoit, Alain Finkel, Francois Laroussinie, Antoine Petit, Laure Petrucci, and Philippe Schnoebelen. *Systems and software verification: model-checking techniques and tools*. Springer Science & Business Media, 2013.

[25] Richard M Bergenstal, William V Tamborlane, Andrew Ahmann, John B Buse, George Dailey, Stephen N Davis, Carol Joyce, Bruce A Perkins, John B Welsh, Steven M Willi, and others. Sensor-Augmented Pump Therapy for A1C Reduction (STAR 3) Study Results from the 6-month continuation phase. *Diabetes Care*, 34(11):2403–2405, 2011.

[26] Christian Berger and Bernhard Rumpe. Autonomous Driving-5 Years after the Urban Challenge: The Anticipatory Vehicle as a Cyber-Physical System. *arXiv preprint arXiv:1409.0413*, 2014.

[27] Richard N Bergman, Y Ziya Ider, CHARLES R Bowden, and Claudio Cobelli. Quantitative estimation of insulin sensitivity. *American Journal of Physiology-Endocrinology And Metabolism*, 236(6):E667, 1979.

[28] Deval Bhatt and L Raymond Reynolds. Keep your hands off my insulin pump! The dilemma of the hospitalized insulin pump patient. *The American journal of medicine*, 2015.

[29] Charles E Billings. *Aviation automation: The search for a human-centered approach*. Taylor & Francis, 1997.

[30] F Bilotta, R Caramia, F P Paoloni, R Delfini, and G Rosa. Safety and efficacy of intensive insulin therapy in critical neurosurgical patients. *Anesthesiology*, 110(3):611–619, 2009.

[31] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[32] Ann Blandford, Richard Butterworth, and Paul Curzon. Models of interactive systems: a case study on programmable user modelling. *International Journal of Human-Computer Studies*, 60(2):149–200, 2004.

[33] Richard J Boland. Control, causality and information system requirements. *Accounting, Organizations and Society*, 4(4):259–272, 1979.

[34] Matthew L Bolton and Ellen J Bass. Formally verifying human–automation interaction as part of a system model: limitations and tradeoffs. *Innovations in systems and software engineering*, 6(3):219–231, 2010.

[35] Matthew L Bolton and Ellen J Bass. Using model checking to explore checklist-guided pilot behavior. *The International Journal of Aviation Psychology*, 22(4):343–366, 2012.

[36] Matthew L Bolton, Ellen J Bass, and Radu I Siminiceanu. Using formal verification to evaluate human-automation interaction: A review. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, 43(3):488–503, 2013.

[37] Matthew L Bolton, Radu Siminiceanu, Ellen J Bass, et al. A systematic approach to model checking human–automation interaction using task analytic models. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(5):961–976, 2011.

[38] Frédéric Bourgault, Nisar Ahmed, Danelle Shah, and Mark Campbell. Probabilistic operator-multiple robot modeling using bayesian network representation. *Proc. GNC, Hilton Head, SC*, 2007.

[39] S S Braithwaite and S Clement. Algorithms for intravenous insulin delivery. *Curr Diabetes Rev*, 4(3):258–268, 2008.

[40] S S Braithwaite, R Edkins, K L Macgregor, E S Sredzienski, M Houston, B Zarzaur, P B Rich, B Benedetto, and E J Rutherford. Performance of a dose-defining insulin infusion protocol among trauma service intensive care unit admissions. *Diabetes Technol Ther*, 8(4):476–488, 2006.

[41] Karel A Brookhuis, Dick De Waard, and Wiel H Janssen. Behavioural impacts of advanced driver assistance systems–an overview. *European Journal of Transport and Infrastructure Research*, 1(3):245–253, 2001.

[42] Ian Brown, Andrew A Adams, et al. The ethical challenges of ubiquitous healthcare. *International Review of Information Ethics*, 8(12):53–60, 2007.

[43] F M Brunkhorst, C Engel, F Bloos, A Meier-Hellmann, M Ragaller, N Weiler, O Moerer, M Gruendling, M Oppert, S Grond, D Olthoff, U Jaschinski, S John, R Rossaint, T Welte, M Schaefer, P Kern, E Kuhnt, M Kiehntopf, C Hartog, C Natanson, M Loeffler, and K Reinhart. Intensive insulin therapy and pentastarch resuscitation in severe sepsis. *N Engl J Med*, 358(2):125–139, 2008.

[44] Alberto Bruno, Dario Gregori, Antonio Caropreso, Fulvio Lazzarato, Michele Petrinco, and Eva Pagano. Normal glucose values are associated with a lower risk of mortality in hospitalized patients. *Diabetes Care*, 31(11):2209–2210, 2008.

[45] Ricky W Butler, Steven P Miller, James N Potts, Victor Carreño, et al. A formal methods approach to the analysis of mode confusion. In *Digital Avionics*

Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE, volume 1, pages C41–1. IEEE, 1998.

[46] Richard Butterworth, Ann Blandford, and David Duke. Demonstrating the cognitive plausibility of interactive system specifications. *Formal Aspects of Computing*, 12(4):237–259, 2000.

[47] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[48] J Creissac Campos and Michael D Harrison. Systematic analysis of control panel interfaces using formal tools. In *Interactive Systems. Design, Specification, and Verification*, pages 72–85. Springer, 2008.

[49] José C Campos and Michael D Harrison. Interaction engineering using the ivy tool. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, pages 35–44. ACM, 2009.

[50] Ewart Carson and Claudio Cobelli. *Modeling methodology for physiology and medicine*. Newnes, 2013.

[51] Antonio Cerone, Peter Lindsay, Simon Connelly, et al. Formal analysis of human-computer interaction using model-checking. In *Software Engineering and Formal Methods, 2005. SEFM 2005. Third IEEE International Conference on*, pages 352–361. IEEE, 2005.

[52] J G Chase, A J Le Compte, F Suhaimi, G M Shaw, A Lynn, J Lin, C G Pretty, N Razak, J D Parente, C E Hann, J C Preiser, and T Desaive. Tight glycemic control in critical care–the leading role of insulin sensitivity and patient variability: a review and model-based analysis. *Comput Methods Programs Biomed*, 102(2):156–171, 2011.

[53] J G Chase, G M Shaw, J Lin, C V Doran, C Hann, T Lotz, G C Wake, and B Broughton. Targeted glycemic reduction in critical care using closed-loop control. *Diabetes Technol Ther*, 7(2):274–282, 2005.

[54] Sanjian Chen, Lu Feng, Michael Rickels, Amy Peleckis, Oleg Sokolsky, and Insup Lee. A data-driven behavior modeling and analysis framework for diabetic patients on insulin pumps. *The IEEE International Conference on Healthcare Informatics 2015 (ICHI 2015)*, 2015.

[55] Sanjian Chen, Matthew O Kelly, James Weimer, Oleg Sokolsky, and Insup Lee. An intraoperative glucose control benchmark for formal verification. *5th IFAC conference on Analysis and Design of Hybrid Systems (ADHS)*, 2015.

[56] Sanjian Chen, James Weimer, Michael Rickels, Amy Peleckis, and Insup Lee. Towards a Model-Based Meal Detector for Type I Diabetics∗. In *the Medical Cyber Physical Systems Workshop, CPS Week, Seattle, WA*, 2015.

[57] Taolue Chen, Marco Diciolla, Marta Kwiatkowska, and Alexandru Mereacre. Quantitative verification of implantable cardiac pacemakers over hybrid heart models. *Information and Computation*, 236:87–101, 2014.

[58] Bruce D Cheson, John M Bennett, Kanti R Rai, Michael R Grever, Neil E Kay, Charles A Schiffer, Martin M Oken, Michael J Keating, David H Boldt, Sanford J Kempin, et al. Guidelines for clinical protocols for chronic lymphocytic leukemia: Recommendations of the national cancer institute-sponsored working group. *American journal of hematology*, 29(3):152–163, 1988.

[59] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 8(2):244–263, 1986.

[60] Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model checking*. MIT press, 1999.

[61] S Clement, S S Braithwaite, M F Magee, A Ahmann, E P Smith, R G Schafer, and I B Hirsch. Management of diabetes and hyperglycemia in hospitals. *Diabetes Care*, 27(2):553–591, 2004.

[62] Claudio Cobelli and Ewart Carson. *Introduction to modeling in physiology and medicine*. Academic Press, 2008.

[63] Claudio Cobelli, Chiara Dalla Man, Giovanni Sparacino, Lalo Magni, Giuseppe De Nicolao, and Boris P Kovatchev. Diabetes: models, signals, and control. *Biomedical Engineering, IEEE Reviews in*, 2:54–96, 2009.

[64] Claudio Cobelli, David Foster, and Gianna Toffolo. *Tracer kinetics in biomedical research*, volume 1. Springer, 2000.

[65] Claudio Cobelli, Giovanni Sparacino, Andrea Caumo, Maria Pia Saccomani, and Gianna Maria Toffolo. Compartmental models of physiologic systems. *The Biomedical Engineering Handbook*, 2:1–11, 2000.

[66] Cobelli, Claudio, Renard, Eric, and Kovatchev, Boris. Artificial pancreas: past, present, future. *Diabetes*, 60(11):2672–2682, 2011.

[67] The Diabetes Control and Complications Trial Research Group. The effect of intensive treatment of diabetes on the development and progression of long-term complications in insulin-dependent diabetes mellitus. *N Engl J Med*, 329(14):977–986, 1993.

[68] Nancy J Cooke. Human factors of remotely operated vehicles. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 166–169. Sage Publications, 2006.

[69] Charles J Coté. Sedation protocolswhy so many variations? *Pediatrics*, 94(3):281–283, 1994.

[70] Philip E Cryer. Hypoglycemia is the limiting factor in the management of diabetes. *Diabetes/metabolism research and reviews*, 15(1):42–46, 1999.

[71] Mary L Cummings and Stephanie Guerlain. Developing operator capacity estimates for supervisory control of autonomous vehicles. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(1):1–15, 2007.

[72] Maria Cvach. Monitor alarm fatigue: an integrative review. *Biomedical Instrumentation & Technology*, 46(4):268–277, 2012.

[73] C Dalla Man, R A Rizza, and C Cobelli. Meal Simulation Model of the Glucose-Insulin System. *IEEE Transactions on Biomedical Engeneering*, page in press, 2007.

[74] C Dalla Man, R A Rizza, and C Cobelli. Meal Simulation Model of the Glucose-Insulin System. *IEEE Transactions on Biomedical Engeneering*, page in press, 2007.

[75] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[76] Goodarz Danaei, Mariel M Finucane, Yuan Lu, Gitanjali M Singh, Melanie J Cowan, Christopher J Paciorek, John K Lin, Farshad Farzadfar, Young-Ho Khang, Gretchen A Stevens, and others. National, regional, and global trends in fasting plasma glucose and diabetes prevalence since 1980: systematic analysis of health examination surveys and epidemiological studies with 370 country-years and 2.7 million participants. *The Lancet*, 378(9785):31–40, 2011.

[77] Eyal Dassau, B Wayne Bequette, Bruce A Buckingham, and Francis J Doyle. Detection of a meal using continuous glucose monitoring implications for an artificial β-cell. *Diabetes care*, 31(2):295–300, 2008.

[78] C De La Rosa Gdel, J H Donado, A H Restrepo, A M Quintero, L G Gonzalez, N E Saldarriaga, M Bedoya, J M Toro, J B Velasquez, J C Valencia, C M Arango, P H Aleman, E M Vasquez, J C Chavarriaga, A Yepes, W Pulido, and C A Cadavid. Strict glycaemic control in patients hospitalised in a mixed medical and surgical intensive care unit: a randomised clinical trial. *Crit Care*, 12(5):R120, 2008.

[79] Patricia Derler, Edward Lee, Alberto Sangiovanni Vincentelli, et al. Modeling cyber–physical systems. *Proceedings of the IEEE*, 100(1):13–28, 2012.

[80] Velin Dimitrov, Vinayak Jagtap, Mitchell Wills, Jeanine Skorinko, and Taskin Padir. A cyber physical system testbed for assistive robotics technologies in the home. In *Advanced Robotics (ICAR), 2015 International Conference on*, pages 323–328. IEEE, 2015.

[81] Alan Dix, Masitah Ghazali, Steve Gill, Joanna Hare, and Devina Ramduny-Ellis. Physigrams: modelling devices for natural interaction. *Formal Aspects of Computing*, 21(6):613–641, 2009.

[82] Stephen R Dixon, Christopher D Wickens, and Jason S McCarley. On the independence of compliance and reliance: Are automation false alarms worse than misses? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(4):564–572, 2007.

[83] L A Dossett, H Cao, N T Mowery, M J Dortch, J M Jr Morris, and A K May. Blood glucose variability is associated with mortality in the surgical intensive care unit. *Am Surg*, 74(8):679–85– discussion 685, 2008.

[84] Katherine Driggs-Campbell, Victor Shia, and Ruzena Bajcsy. Improved driver modeling for human-in-the-loop vehicular control. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1654–1661. IEEE, 2015.

[85] Ismael Duque, Kerstin Dautenhahn, Kheng Lee Koay, Ian Willcock, and Bruce Christianson. Knowledge-driven user activity recognition for a Smart House. Development and validation of a generic and low-cost, resource-efficient system. In *The Sixth International conference on advances in computer–human interactions (ACHI 2013), Nice, France*, 2013.

[86] Matthew B Dwyer, Oksana Tkachuk, Willem Visser, et al. Analyzing interaction orderings with model checking. In *Proceedings of the 19th IEEE international conference on Automated software engineering*, pages 154–163. IEEE Computer Society, 2004.

[87] M Egi, R Bellomo, E Stachowski, C J French, and G Hart. Variability of blood glucose concentration and short-term mortality in critically ill patients. *Anesthesiology*, 105(2):244–252, 2006.

[88] Christian Ellingsen, Eyal Dassau, Howard Zisser, Benyamin Grosman, Matthew W Percival, Lois Jovanovic, and Francis J Doyle. Safety constraints in an artificial pancreatic β cell: an implementation of model predictive control with insulin on board. *Journal of diabetes science and technology*, 3(3):536–544, 2009.

[89] E Wesley Ely, Maureen O Meade, Edward F Haponik, Marin H Kollef, Deborah J Cook, Gordon H Guyatt, and James K Stoller. Mechanical ventilator weaning protocols driven by nonphysician health-care professionals: evidence-based clinical practice guidelines. *CHEST Journal*, 120(6_suppl):454S–463S, 2001.

[90] Randall W Engle. Working memory capacity as executive attention. *Current directions in psychological science*, 11(1):19–23, 2002.

[91] Mitra M Fatourechi, Yogish C Kudva, M Hassan Murad, Mohamed B Elamin, Claudia C Tabini, and Victor M Montori. Hypoglycemia with intensive insulin therapy: a systematic review and meta-analyses of randomized trials of continuous subcutaneous insulin infusion versus multiple daily injections. *The Journal of Clinical Endocrinology & Metabolism*, 94(3):729–740, 2009.

[92] S Finfer, D R Chittock, S Y Su, D Blair, D Foster, V Dhingra, R Bellomo, D Cook, P Dodek, W R Henderson, Hebert, P. C., S Heritier, D K Heyland, C McArthur, E McDonald, I Mitchell, J A Myburgh, R Norton, J Potter, B G Robinson, and J J Ronco. Intensive versus conventional glucose control in critically ill patients. *N Engl J Med*, 360(13):1283–1297, 2009.

[93] S Finfer, B Liu, D R Chittock, R Norton, J A Myburgh, C McArthur, I Mitchell, D Foster, V Dhingra, W R Henderson, J J Ronco, R Bellomo, D Cook, E McDonald, P Dodek, Hebert, P. C., D K Heyland, and B G Robinson. Hypoglycemia and risk of death in critically ill patients. *N Engl J Med*, 367(12):1108–1118, 2012.

[94] Donald L Fisher, Nancy E Laurie, Robert Glaser, Karen Connerney, Alexander Pollatsek, Susan A Duffy, and John Brock. Use of a fixed-base driving simulator to evaluate the effects of experience and pc-based risk awareness training on drivers' decisions. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 44(2):287–302, 2002.

[95] Centers for Disease Control, Prevention, et al. National diabetes statistics report: estimates of diabetes and its burden in the united states, 2014. *Atlanta, ga: US Department of health and human services*, 2014.

[96] Ken Funk, Beth Lyall, Jennifer Wilson, Rebekah Vint, Mary Niemczyk, Candy Suroteguh, and Griffith Owen. Flight deck automation issues. *The International Journal of Aviation Psychology*, 9(2):109–123, 1999.

[97] Sicun Gao, Soonho Kong, and Edmund M Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *Automated Deduction–CADE-24*, pages 208–214. Springer, 2013.

[98] Sicun Gao, Soonho Kong, and Edmund M Clarke. Satisfiability modulo odes. In *Formal Methods in Computer-Aided Design (FMCAD), 2013*, pages 105–112. IEEE, 2013.

[99] Rachel Gillis, Cesar C Palerm, Howard Zisser, Lois Jovanovic, Dale E Seborg, and Francis J Doyle. Glucose estimation and prediction through meal responses using ambulatory subject data for advisory mode model predictive control. *Journal of diabetes science and technology*, 1(6):825–833, 2007.

[100] Keith Godfrey. Compartmental models and their application. In *Compartmental models and their application*. Academic Press, 1983.

[101] E W Gregg, Q Gu, Y J Cheng, K M Narayan, and C C Cowie. Mortality trends in men and women with diabetes, 1971 to 2000. *Ann Intern Med*, 147(3):149–155, 2007.

[102] UK Prospective Diabetes Study UKPDS Group. Intensive blood-glucose control with sulphonylureas or insulin compared with conventional treatment and risk of complications in patients with type 2 diabetes (UKPDS 33). *Lancet*, 352(9131):837–853, 1998.

[103] George Grunberger, Jill M Abelseth, Timothy S Bailey, Bruce W Bode, Yehuda Handelsman, Richard Hellman, Lois Jovanovic, Wendy S Lane, Philip Raskin, William V Tamborlane, and others. Consensus statement by the American

Association of Clinical Endocrinologists/American College of Endocrinology insulin pump management task force. *Endocrine Practice*, 20(5):463–489, 2014.

[104] K Gu, C C Cowie, and M I Harris. Mortality in adults with and without diabetes in a national cohort of the U.S. population, 1971-1993. *Diabetes Care*, 21(7):1138–1145, 1998.

[105] David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.

[106] Maureen I Harris, Wilbur C Hadden, William C Knowler, and Peter H Bennett. Prevalence of diabetes and impaired glucose tolerance and plasma glucose levels in US population aged 20–74 yr. *Diabetes*, 36(4):523–534, 1987.

[107] John A Hartigan and Manchek A Wong. Algorithm AS 136: A k-means clustering algorithm. *Applied statistics*, pages 100–108, 1979.

[108] H Rex Hartson, Antonio C Siochi, and Deborah Hix. The uan: A user-oriented representation for direct manipulation interface designs. *ACM Transactions on Information Systems (TOIS)*, 8(3):181–203, 1990.

[109] Rebecca A Harvey, Eyal Dassau, Howard Zisser, Dale E Seborg, and Francis J Doyle. Design of the Glucose Rate Increase Detector A Meal Detection Module for the Health Monitoring System. *Journal of diabetes science and technology*, page 1932296814523881, 2014.

[110] Michele Heisler, Sandeep Vijan, Fatima Makki, and John D Piette. Diabetes control with reciprocal peer support versus nurse care management: a randomized trial. *Annals of internal medicine*, 153(8):507–515, 2010.

[111] Thomas A Henzinger. *The theory of hybrid automata*. Springer, 2000.

[112] Thomas A Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HyTech: A model checker for hybrid systems. In *Computer aided verification*, pages 460–463. Springer, 1997.

[113] Pei-Cheng Hii and Wan-Young Chung. A comprehensive ubiquitous healthcare solution on an android mobile device. *Sensors*, 11(7):6799–6815, 2011.

[114] M Hoekstra, M Vogelzang, E Verbitskiy, and M W Nijsten. Health technology assessment review: Computerized glucose regulation in the intensive care unit–how to create artificial control. *Crit Care*, 13(5):223, 2009.

[115] Gerard J Holzmann. *The SPIN model checker: Primer and reference manual*, volume 1003. Addison-Wesley Reading, 2004.

[116] M Horibe, B G Nair, G Yurina, M B Neradilek, and I Rozet. A novel computerized fading memory algorithm for glycemic control in postoperative surgical patients. *Anesth Analg*, 115(3):580–587, 2012.

[117] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[118] Roman Hovorka, Ludovic J Chassin, Martin Ellmerer, Johannes Plank, and Malgorzata E Wilinska. A simulation model of glucose regulation in the critically ill. *Physiological measurement*, 29(8):959, 2008.

[119] F B Hu, M J Stampfer, C G Solomon, S Liu, W C Willett, F E Speizer, D M Nathan, and J E Manson. The impact of diabetes mellitus on mortality from all causes and coronary heart disease in women: 20 years of follow-up. *Arch Intern Med*, 161(14):1717–1723, 2001.

[120] David Hughes and Michael A Dornheim. Accidents direct focus on cockpit automation. *Aviation Week and Space Technology*, 142(5):52–54, 1995.

[121] Inseok Hwang and Chze Eng Seah. Intent-based probabilistic conflict detection for the next generation air transportation system. *Proceedings of the IEEE*, 96(12):2040–2059, 2008.

[122] Toshiyuki Inagaki, Makoto Itoh, and Yoshitomo Nagai. Driver support functions under resource-limited situations. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 51, pages 176–180. Sage Publications, 2007.

[123] Radoslav Ivanov, James Weimer, Allan Simpao, Mohamed Rehman, and Insup Lee. Early detection of critical pulmonary shunts in infants. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, pages 110–119. ACM, 2015.

[124] JA Jacquez. Compartmental analysis in biology and medicine. *BioMedware*, 1996.

[125] Alejandro Jaimes and Nicu Sebe. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, 108(1):116–134, 2007.

[126] R G Janes and J O Osburn. The analysis of glucose measurements by computer simulation. *J Physiol*, 181(1):59–67, 1965.

[127] Denis Javaux. A method for predicting errors when interacting with finite state systems. how implicit learning shapes the user's knowledge of a system. *Reliability Engineering & System Safety*, 75(2):147–165, 2002.

[128] K Jeitler, K Horvath, A Berghold, T W Gratzer, K Neeser, T R Pieber, and A Siebenhofer. Continuous subcutaneous insulin infusion versus multiple daily insulin injections in patients with diabetes mellitus: systematic review and meta-analysis. *Diabetologia*, 51(6):941–951, 2008.

[129] Zhihao Jiang, Miroslav Pajic, and Rahul Mangharam. Model-based closed-loop testing of implantable pacemakers. In *Proceedings of the 2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*, pages 131–140. IEEE Computer Society, 2011.

[130] Zhihao Jiang, Miroslav Pajic, Salar Moarref, Rajeev Alur, and Rahul Mangharam. Modeling and verification of a dual chamber implantable pacemaker. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 188–203. Springer, 2012.

[131] J Joe, John O'Hara, H Medema, and J Oxstrand. Identifying Requirements for Effective Human-Automation Teamwork. In *Proceedings of the 12th International Conference on Probabilistic Safety Assessment and Management*, 2014.

[132] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[133] Emil Jovanov and Aleksandar Milenkovic. Body area networks for ubiquitous healthcare applications: opportunities and challenges. *Journal of medical systems*, 35(5):1245–1254, 2011.

[134] Joonyoung Jung, Kiryong Ha, Jeonwoo Lee, Youngsung Kim, and Daeyoung Kim. Wireless body area network in a ubiquitous healthcare system for physiological signal monitoring and health consulting. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 1(1):47–54, 2008.

[135] Aaron V Kaplan, Donald S Baim, John J Smith, David A Feigal, Michael Simons, David Jefferys, Thomas J Fogarty, Richard E Kuntz, and Martin B Leon. Medical device development from prototype to regulatory approval. *Circulation*, 109(25):3068–3072, 2004.

[136] Fakhreddine Karray, Milad Alemzadeh, Jamil Abou Saleh, and Mo Nours Arab. Human-computer interaction: Overview on state of the art. *Citeseer*, 2008.

[137] Madhav A Karunakar and Kurtis S Staples. Does stress-induced hyperglycemia increase the risk of perioperative infectious complications in orthopaedic trauma patients? *Journal of orthopaedic trauma*, 24(12):752–756, 2010.

[138] Arie Katz, Sridhar S Nambi, Kieren Mather, Alain D Baron, Dean A Follmann, Gail Sullivan, and Michael J Quon. Quantitative insulin sensitivity check index: a simple, accurate method for assessing insulin sensitivity in humans. *The Journal of Clinical Endocrinology & Metabolism*, 85(7):2402–2410, 2000.

[139] Isam A Kaysi and Ali S Abbany. Modeling aggressive driver behavior at unsignalized intersections. *Accident Analysis & Prevention*, 39(4):671–678, 2007.

[140] Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.

[141] Sarfraz Khurshid, Corina S Pasareanu, and Willem Visser. Generalized symbolic execution for model checking and testing. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 553–568. Springer, 2003.

[142] Kyoung-Dae Kim and Panganamala R Kumar. Cyber–physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(Special Centennial Issue):1287–1308, 2012.

[143] Barry Kirwan and Les K Ainsworth. *A guide to task analysis: the task analysis working group*. CRC press, 1992.

[144] Ker-I Ko. *Complexity theory of real functions*. Birkhauser Boston Inc., 1991.

[145] Tetsuro Kobayashi, Shinji Sawano, Tokuji Itoh, Kinori Kosaka, Hiroki Hirayama, and Yasuji Kasuya. The pharmacokinetics of insulin after continuous subcutaneous infusion or bolus subcutaneous injection in diabetic patients. *Diabetes*, 32(4):331–336, 1983.

[146] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[147] Benjamin Kohl, Sanjian Chen, Margaret Mullen-Fortino, Insup Lee, et al. Evaluation and enhancement of an intraoperative insulin infusion protocol via in-silico simulation. In *Healthcare Informatics (ICHI), 2013 IEEE International Conference on*, pages 307–316. IEEE, 2013.

[148] Linda T Kohn, Janet M Corrigan, Molla S Donaldson, et al. *To err is human:: building a Safer Health System*, volume 6. National Academies Press, 2000.

[149] Soonho Kong, Sicun Gao, Wei Chen, and Edmund M Clarke. dReach: Delta-Reachability Analysis for Hybrid Systems. In *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, pages 200–205, 2015.

[150] B P Kovatchev and W L Clarke. Continuous glucose monitoring reduces risks for hypo- and hyperglycemia and glucose variability in diabetes. *Diabetes*, 56 (suppl 1):0086OR, 2007.

[151] Boris P Kovatchev, Marc Breton, Chiara Dalla Man, and Claudio Cobelli. In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes. *Journal of diabetes science and technology*, 3(1):44–55, 2009.

[152] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering,

and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1, 2009.

[153] J S Krinsley. Association between hyperglycemia and increased hospital mortality in a heterogeneous population of critically ill patients. *Mayo Clin Proc*, 78(12):1471–1478, 2003.

[154] J S Krinsley. Glycemic variability: a strong independent predictor of mortality in critically ill patients. *Crit Care Med*, 36(11):3008–3013, 2008.

[155] Eray Kulcu, Janet A Tamada, Gerard Reach, Russell O Potts, and Matthew J Lesho. Physiological differences between interstitial glucose and blood glucose measured in human subjects. *Diabetes care*, 26(8):2405–2409, 2003.

[156] Santosh Kumar, Wendy Nilsen, Misha Pavel, and Mani Srivastava. Mobile health: Revolutionizing healthcare through transdisciplinary research. *Computer*, pages 28–35, 2013.

[157] M Kwiatkowska, G Norman, and D Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *CAV*, pages 585–591. Springer, 2011.

[158] Markku Laakso. How good a marker is insulin level for insulin resistance? *American Journal of Epidemiology*, 137(9):959–965, 1993.

[159] Janice Langan-Fox, James M Canty, and Michael J Sankey. Human–automation teams and adaptable control for future air traffic management. *International Journal of Industrial Ergonomics*, 39(5):894–903, 2009.

[160] Kim G Larsen, Paul Pettersson, and Wang Yi. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1):134–152, 1997.

[161] Harold L Lazar, Stuart R Chipkin, Carmel A Fitzgerald, Yusheng Bao, Howard Cabral, and Carl S Apstein. Tight glycemic control in diabetic coronary artery

bypass graft patients improves perioperative outcomes and decreases recurrent ischemic events. *Circulation*, 109(12):1497–1502, 2004.

[162] A J Le Compte, C G Pretty, J Lin, G M Shaw, A Lynn, and J G Chase. Impact of variation in patient response on model-based control of glycaemia in critically ill patients. *Comput Methods Programs Biomed*, 2011.

[163] Fred D Ledley. Clinical considerations in the design of protocols for somatic gene therapy. *Human gene therapy*, 2(1):77–83, 1991.

[164] Anthony Lee, Badia Faddoul, Azizeh Sowan, Karen L Johnson, Kristi D Silver, and Vinay Vaidya. Computerisation of a paper-based intravenous insulin protocol reduces errors in a prospective crossover simulated tight glycaemic control study. *Intensive and Critical Care Nursing*, 26(3):161–168, 2010.

[165] Edward Lee et al. Cyber physical systems: Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369. IEEE, 2008.

[166] Edward A Lee. Cyber-physical systems-are computing foundations adequate. In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, volume 2. Citeseer, 2006.

[167] Hak Jong Lee, Sun Hee Lee, Kyoo-Seob Ha, Hak Chul Jang, Woo-Young Chung, Ju Young Kim, Yoon-Seok Chang, and Dong Hyun Yoo. Ubiquitous healthcare service using zigbee and mobile phone for elderly patients. *International journal of medical informatics*, 78(3):193–198, 2009.

[168] Hyunjin Lee and B Wayne Bequette. A closed-loop artificial pancreas based on model predictive control: Human-friendly identification and automatic meal disturbance rejection. *Biomedical Signal Processing and Control*, 4(4):347–354, 2009.

[169] Insup Lee, Oleg Sokolsky, Sanjian Chen, John Hatcliff, Eunkyoung Jee, Baek-Gyu Kim, Andrew King, Margaret Mullen-Fortino, Soojin Park, Alexander Roederer, and Krishna K Venkatasubramanian. Challenges and Research Directions in Medical Cyber-Physical Systems. *Proceedings of the IEEE*, 100(1):75–90, January 2012.

[170] J C Lee, M Kim, K R Choi, T J Oh, M Y Kim, Y M Cho, K Kim, H C Kim, and S Kim. In silico evaluation of glucose control protocols for critically ill patients. *IEEE Trans Biomed Eng*, 59(1):54–57, 2012.

[171] John Lee and Neville Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35(10):1243–1270, 1992.

[172] John D Lee and Neville Moray. Trust, self-confidence, and operators' adaptation to automation. *International journal of human-computer studies*, 40(1):153–184, 1994.

[173] G Leibowitz, E Raizman, M Brezis, B Glaser, I Raz, and O Shapira. Effects of moderate intensity glycemic control after cardiac surgery. *Ann Thorac Surg*, 90(6):1825–1832, 2010.

[174] Nancy Leveson, L Denise Pinnel, Sean David Sandys, Shuichi Koga, and Jon Damon Reese. Analyzing software specifications for mode confusion potential. In *Proceedings of a Workshop on Human Error and System Development*, pages 132–146, 1997.

[175] Nancy G Leveson and Clark S Turner. An investigation of the therac-25 accidents. *Computer*, 26(7):18–41, 1993.

[176] C S Levetan, M Passaro, K Jablonski, M Kass, and R E Ratner. Unrecognized diabetes among hospitalized patients. *Diabetes Care*, 21(2):246–249, 1998.

[177] Soo Lim, Seon Mee Kang, Hayley Shin, Hak Jong Lee, Ji Won Yoon, Sung Hoon Yu, So-Youn Kim, Soo Young Yoo, Hye Seung Jung, Kyong Soo Park, et al. Improved glycemic control without hypoglycemia in elderly diabetic patients using the ubiquitous healthcare service, a new medical information system. *Diabetes care*, 34(2):308–313, 2011.

[178] Jessica Lin, Dominic Lee, J Geoffrey Chase, Geoffrey M Shaw, Aaron Le Compte, Thomas Lotz, Jason Wong, Timothy Lonergan, and Christopher E Hann. Stochastic modelling of insulin sensitivity and adaptive glycemic control for critical care. *Computer methods and programs in biomedicine*, 89(2):141–152, 2008.

[179] Peter Lindsay and Simon Connelly. Modelling erroneous operator behaviours for an air-traffic control task. In *Australian Computer Science Communications*, volume 24, pages 43–54. Australian Computer Society, Inc., 2002.

[180] Angela KM Lipshutz and Michael A Gropper. Perioperative glycemic controlan evidence-based review. *The Journal of the American Society of Anesthesiologists*, 110(2):408–421, 2009.

[181] Benny PL Lo, Surapa Thiemjarus, Rachel King, and Guang-Zhong Yang. *Body sensor network–a wireless sensor platform for pervasive healthcare monitoring.* na, 2005.

[182] Timothy Lonergan, Aaron Le Compte, Mike Willacy, J Geoffrey Chase, Geoffrey M Shaw, Xing-Wei Wong, Thomas Lotz, Jessica Lin, and Christopher E Hann. A simple insulin-nutrition protocol for tight glycemic control in critical illness: development and protocol comparison. *Diabetes technology & therapeutics*, 8(2):191–206, 2006.

[183] Thomas F Lotz, J Geoffrey Chase, Kirsten A McAuley, Geoffrey M Shaw, Xing-Wei Wong, Jessica Lin, Aaron LeCompte, Christopher E Hann, and Jim I

Mann. Monte carlo analysis of a new model-based method for insulin sensitivity testing. *Computer methods and programs in biomedicine*, 89(3):215–225, 2008.

[184] Rogelio Lozano. *Unmanned aerial vehicles: Embedded control*. John Wiley & Sons, 2013.

[185] Jiakang Lu, Tamim Sookoor, Vijay Srinivasan, Ge Gao, Brian Holben, John Stankovic, Eric Field, and Kamin Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 211–224. ACM, 2010.

[186] Joseph B Lyons, Kolina S Koltai, Nhut T Ho, Walter B Johnson, David E Smith, and R Jay Shively. Engineering trust in complex automated systems. *Ergonomics in Design: The Quarterly of Human Factors Applications*, 24(1):13–17, 2016.

[187] L Magni, DM Raimondo, Ch Dalla Man, G De Nicolao, B Kovatchev, and C Cobelli. Model predictive control of glucose concentration in type i diabetic patients: An in silico trial. *Biomedical Signal Processing and Control*, 4(4):338–346, 2009.

[188] Lalo Magni, Davide M Raimondo, Luca Bossi, Chiara Dalla Man, Giuseppe De Nicolao, Boris Kovatchev, and Claudio Cobelli. Model predictive control of type 1 diabetes: an in silico trial. *Journal of diabetes science and technology*, 1(6):804–812, 2007.

[189] K Malmberg. Prospective randomised study of intensive insulin treatment on long term survival after acute myocardial infarction in patients with diabetes mellitus. DIGAMI (Diabetes Mellitus, Insulin Glucose Infusion in Acute Myocardial Infarction) Study Group. *Bmj*, 314(7093):1512–1515, 1997.

[190] K Malmberg, A Norhammar, H Wedel, and L Ryden. Glycometabolic state at admission: important risk marker of mortality in conventionally treated patients with diabetes mellitus and acute myocardial infarction: long-term results from the Diabetes and Insulin-Glucose Infusion in Acute Myocardial Infarction (DIGAMI) study. *Circulation*, 99(20):2626–2632, 1999.

[191] Vasilis Z Marmarelis. *Nonlinear dynamic modeling of physiological systems*, volume 10. John Wiley & Sons, 2004.

[192] Paolo Masci, Paul Curzon, Ann Blandford, and Dominic Furniss. Modelling distributed cognition systems in pvs. *Electronic Communications of the EASST*, 45, 2011.

[193] Finlay A McAlister, Sumit R Majumdar, Sandra Blitz, Brian H Rowe, Jacques Romney, and Thomas J Marrie. The relation between hyperglycemia and outcomes in 2,471 patients admitted to the hospital with community-acquired pneumonia. *Diabetes Care*, 28(4):810–815, 2005.

[194] Karen C McCowen, Atul Malhotra, and Bruce R Bistrian. Stress-induced hyperglycemia. *Critical care clinics*, 17(1):107–124, 2001.

[195] Matt McFarland. Tesla's autopilot probed by government after crash kills driver. *URL http://money.cnn.com/2016/06/30/technology/tesla-autopilot-death/*, July 2016. [Online; accessed July-10-2016].

[196] Kenneth L McMillan. Using unfoldings to avoid the state explosion problem in the verification of asynchronous circuits. In *Computer Aided Verification*, pages 164–177. Springer, 1993.

[197] Jerry Meece. The Artificial Pancreas Where We Are, Where We're Going. *AADE in Practice*, 3(2):42–44, 2015.

[198] Sofie Meijering, Anouk M Corstjens, Jaap E Tulleken, John HJM Meertens, Jan G Zijlstra, and Jack JM Ligtenberg. Towards a feasible algorithm for tight glycaemic control in critically ill patients: a systematic review of the literature. *Critical Care*, 10(1):R19, 2006.

[199] John A Michon. A critical view of driver behavior models: what do we know, what should we do? In *Human behavior and traffic safety*, pages 485–524. Springer, 1985.

[200] Marie L Misso, Kristine J Egberts, Matthew Page, Denise O'Connor, and Jonathan Shaw. Continuous subcutaneous insulin infusion (CSII) versus multiple insulin injections for type 1 diabetes mellitus. *The Cochrane Library*, 2010.

[201] Christine M Mitchell and Richard A Miller. A discrete control model of operator function: A methodology for information display design. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(3):343–357, 1986.

[202] Chiyomi Miyajima, Yoshihiro Nishiwaki, Koji Ozawa, Toshihiro Wakita, Katsunobu Itou, Kazuya Takeda, and Fumitada Itakura. Driver modeling based on driving behavior and its evaluation in driver identification. *Proceedings of the IEEE*, 95(2):427–437, 2007.

[203] Kathleen L Mosier and Ute M Fischer. Judgment and decision making by individuals and teams: issues, models, and applications. *Reviews of Human factors and Ergonomics*, 6(1):198–256, 2010.

[204] G P Moustris, S C Hiridis, K M Deliparaschos, and K M Konstantinidis. Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 7(4):375–392, 2011.

[205] Bonnie M Muir. Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies*, 27(5):527–539, 1987.

[206] Sirajum Munir, John A Stankovic, Chieh-Jan Mike Liang, and Shan Lin. Cyber physical system challenges for human-in-the-loop control. In *Presented as part of the 8th International Workshop on Feedback Computing, Berkeley, CA*, 2013.

[207] M H Murad, J A Coburn, F Coto-Yglesias, S Dzyubak, A Hazem, M A Lane, L J Prokop, and V M Montori. Glycemic control in non-critically ill hospitalized patients: a systematic review and meta-analysis. *J Clin Endocrinol Metab*, 97(1):49–58, 2012.

[208] Kazunari Nawa, Naiwala P Chandrasiri, Tadashi Yanagihara, and Kentaro Oguchi. Cyber physical system for vehicle application. *Transactions of the Institute of Measurement and Control*, 36(7):898–905, 2014.

[209] P G Noordzij, E Boersma, F Schreiner, M D Kertai, H H Feringa, M Dunkelgrun, J J Bax, J Klein, and D Poldermans. Increased preoperative glucose levels are associated with perioperative mortality in patients undergoing noncardiac, nonvascular surgery. *Eur J Endocrinol*, 156(1):137–142, 2007.

[210] D A Norman. The 'Problem' with Automation: Inappropriate Feedback and Interaction, not 'Over-Automation'. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 327(1241):585–593, 1990.

[211] Donald A Norman. Some observations on mental models. *Mental models*, 7(112):7–14, 1983.

[212] Gianluca Nucci and Claudio Cobelli. Models of subcutaneous insulin kinetics. a critical review. *Computer methods and programs in biomedicine*, 62(3):249–257, 2000.

[213] Katsuhiko Ogata. *Discrete-time control systems*, volume 2. Prentice Hall Englewood Cliffs, NJ, 1995.

[214] John M O'Hara, J Higgins, W Brown, Robert Fink, J Persensky, P Lewis, J Kramer, A Szabo, and M Boggi. *Human factors considerations with respect to emerging technology in nuclear power plants*. US Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 2008.

[215] Kumiko Ohashi, Olivia Dalleur, Patricia C Dykes, and David W Bates. Benefits and risks of using smart pumps to reduce medication error rates: a systematic review. *Drug safety*, 37(12):1011–1020, 2014.

[216] C Pachler, J Plank, H Weinhandl, L J Chassin, M E Wilinska, R Kulnik, P Kaufmann, K H Smolle, E Pilger, T R Pieber, M Ellmerer, and R Hovorka. Tight glycaemic control by an automated algorithm with time-variant sampling in medical ICU patients. *Intensive Care Med*, 34(7):1224–1230, 2008.

[217] Philippe A Palanque, Rémi Bastide, and Valérie Sengès. Validating interactive system design through the verification of formal task and system models. In *EHCI*, pages 189–212, 1995.

[218] Raja Parasuraman, Robert Molloy, and Indramani L Singh. Performance consequences of automation-induced'complacency'. *The International Journal of Aviation Psychology*, 3(1):1–23, 1993.

[219] Raja Parasuraman and Victor Riley. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors*, 39(2):230–253, 1997.

[220] David L Parnas. On the use of transition diagrams in the design of a user interface for an interactive computer system. In *Proceedings of the 1969 24th national conference*, pages 379–385. ACM, 1969.

[221] Stephen D Patek, B Wayne Bequette, Marc Breton, Bruce A Buckingham, Eyal Dassau, Francis J Doyle, John Lum, Lalo Magni, and Howard Zisser. In silico preclinical trials: methodology and engineering guide to closed-loop control in type 1 diabetes mellitus. *Journal of diabetes science and technology*, 3(2):269–282, 2009.

[222] Stephen D Patek, Sanjian Chen, Patrick Keith-Hynes, and Insup Lee. Distributed aspects of the artificial pancreas. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 543–550. IEEE, 2013.

[223] Fabio Paterno. Formal reasoning about dialogue properties with automatic support. *Interacting with computers*, 9(2):173–196, 1997.

[224] Fabio Paternò, Cristiano Mancini, and Silvia Meniconi. Concurtasktrees: A diagrammatic notation for specifying task models. In *Human-Computer Interaction INTERACT97*, pages 362–369. Springer, 1997.

[225] M W Percival, Y Wang, B Grosman, E Dassau, H Zisser, L Jovanovic, and F J 3rd Doyle. Development of a multi-parametric model predictive control algorithm for insulin delivery in type 1 diabetes mellitus using clinical parameters. *J Process Control*, 21(3):391–404, 2011.

[226] Charles Perrow. *Normal accidents: Living with high risk technologies*. Princeton University Press, 2011.

[227] J C Pickup and A J Sutton. Severe hypoglycaemia and glycaemic control in Type 1 diabetes: meta-analysis of multiple daily insulin injections compared with continuous subcutaneous insulin infusion. *Diabetic Medicine*, 25(7):765–774, 2008.

[228] Zachary D Post, Camilo Restrepo, Lauren K Kahl, Tim van de Leur, James J Purtill, and William J Hozack. A prospective evaluation of 2 different pain

management protocols for total hip arthroplasty. *The Journal of arthroplasty*, 25(3):410–415, 2010.

[229] J C Preiser, P Devos, S Ruiz-Santana, C Melot, D Annane, J Groeneveld, G Iapichino, X Leverve, G Nitenberg, P Singer, J Wernerman, M Joannidis, A Stecher, and R Chiolero. A prospective randomised multi-centre controlled trial on tight glucose control by intensive insulin therapy in adult intensive care units: the Glucontrol study. *Intensive Care Med*, 35(10):1738–1748, 2009.

[230] Alberto Alessandro Angelo Puggelli. *Formal Techniques for the Verification and Optimal Control of Probabilistic Systems in the Presence of Modeling Uncertainties*. PhD thesis, University of California, Berkeley, 2014.

[231] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[232] Ragunathan Raj Rajkumar, Insup Lee, Lui Sha, and John Stankovic. Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th Design Automation Conference*, pages 731–736. ACM, 2010.

[233] James Reason. *Human error*. Cambridge university press, 1990.

[234] Ulrich Reiser, Theo Jacobs, Georg Arbeiter, Christopher Parlitz, and Kerstin Dautenhahn. Care-O-bot® 3–Vision of a robot butler. *Your virtual butler. Springer*, pages 97–116, 2013.

[235] Justin E Richards, Rondi M Kauffmann, William T Obremskey, and Addison K May. Stress-induced hyperglycemia as a risk factor for surgical-site infection in non-diabetic orthopaedic trauma patients admitted to the intensive care unit. *Journal of orthopaedic trauma*, 27(1):16, 2013.

[236] Alexander Roederer, James Weimer, Joseph DiMartino, Jacob Gutsche, and Insup Lee. Robust monitoring of hypovolemia in intensive care patients using

photoplethysmogram signals. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 1504–1507. IEEE, 2015.

[237] Edmond A Ryan, Tami Shandro, Kristy Green, Breay W Paty, Peter A Senior, David Bigam, AM James Shapiro, and Marie-Christine Vantyghem. Assessment of the severity of hypoglycemia and glycemic lability in type 1 diabetic subjects undergoing islet transplantation. *Diabetes*, 53(4):955–962, 2004.

[238] Dorsa Sadigh, Katherine Driggs-Campbell, Alberto Puggelli, Wenchao Li, Victor Shia, Ruzena Bajcsy, Alberto L Sangiovanni-Vincentelli, S Shankar Sastry, and Sanjit A Seshia. Data-driven probabilistic modeling and verification of human driver behavior. *Formal Verification and Modeling in Human-Machine Systems*, 2014.

[239] Eduardo Salas, Florian Jentsch, and Dan Maurino. *Human factors in aviation*. Academic Press, 2010.

[240] Nadine B Sarter and David D Woods. How in the World Did We Ever Get into That Mode? Mode Error and Awareness in Supervisory Control. *Human Factors*, 37(1):5–19, 1995.

[241] Sarter, N. Investigating Mode Errors on Automated Flight Decks: Illustrating the Problem-Driven, Cumulative, and Interdisciplinary Nature of Human Factors Research. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):506–510, June 2008.

[242] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[243] Richard M Satava. Virtual reality surgical simulator. *Surgical endoscopy*, 7(3):203–205, 1993.

[244] Louis L Scharf. *Statistical signal processing*, volume 98. Addison-Wesley Reading, MA, 1991.

[245] Gunar Schirner, Deniz Erdogmus, Kaushik R Chowdhury, and Taskin Padir. The Future of Human-in-the-Loop Cyber-Physical Systems. *IEEE Computer ()*, 46(1):36–45, 2013.

[246] Bobbie D Seppelt and John D Lee. Making adaptive cruise control (acc) limits visible. *International Journal of Human-Computer Studies*, 65(3):192–205, 2007.

[247] B Shashaj, E Busetto, and N Sulli. Benefits of a bolus calculator in pre-and postprandial glycaemic control and meal flexibility of paediatric patients using continuous subcutaneous insulin infusion (csii). *Diabetic Medicine*, 25(9):1036–1042, 2008.

[248] Victor Shia, Yiqi Gao, Ramanarayan Vasudevan, Katherine Driggs Campbell, Tao Lin, Francesco Borrelli, Ruzena Bajcsy, and others. Semiautonomous vehicular control using driver modeling. *Intelligent Transportation Systems, IEEE Transactions on*, 15(6):2696–2709, 2014.

[249] R Shulman, S J Finney, C O'Sullivan, P A Glynne, and R Greene. Tight glycaemic control: a prospective observational study of a computerised decision-supported intensive insulin therapy protocol. *Crit Care*, 11(4):R75, 2007.

[250] Maarten Sierhuis and William J Clancey. Modeling and simulating practices, a work method for work systems design. *Intelligent Systems, IEEE*, 17(5):32–41, 2002.

[251] Sathya S Silva and R John Hansman. Divergence between flight crew mental model and aircraft system state in auto-throttle mode confusion accident and incident cases. *Journal of Cognitive Engineering and Decision Making*, page 1555343415597344, 2015.

[252] StatSoft. Finding the right number of clusters in k-means and em clustering: v-fold cross-validation. *Electronic Statistics Textbook*, 2010.

[253] C Stettler, S Allemann, P Juni, C A Cull, R R Holman, M Egger, S Krahenbuhl, and P Diem. Glycemic control and macrovascular disease in types 1 and 2 diabetes mellitus: Meta-analysis of randomized trials. *Am Heart J*, 152(1):27–38, 2006.

[254] Richard Stocker. *Towards the formal verification of human-agent-robot teamwork*. PhD thesis, University of Liverpool, 2013.

[255] Richard Stocker, Louise Dennis, Clare Dixon, and Michael Fisher. Verifying brahms human-robot teamwork models. In *Logics in Artificial Intelligence*, pages 385–397. Springer, 2012.

[256] Balachundhar Subramaniam, Peter Panzica, Victor Novack, Feroze Mahmood, Robina Matyal, John Mitchell, Eswar Sundar, Ruma Bose, Frank Pomposelli, Judy Kersten, et al. Continuous perioperative insulin infusion decreases major cardiovascular events in patients undergoing vascular surgery: a prospective, randomized trial. *Anesthesiology*, 110(5):970, 2009.

[257] United States. Federal Aviation Administration. Human Factors Team. *Federal Aviation Administration Human Factors Team Report on the Interfaces Between Flightcrews and Modern Flight Deck Systems*. Federal Aviation Administration, 1996.

[258] Miriam M Treggiari, Veena Karir, N David Yanez, Noel S Weiss, Stephen Daniel, and Steven A Deem. Intensive insulin therapy and mortality in critically ill patients. *Critical Care*, 12(1):R29, 2008.

[259] G E Umpierrez, R Hellman, M T Korytkowski, M Kosiborod, G A Maynard, V M Montori, J J Seley, and G Van den Berghe. Management of hyperglycemia

in hospitalized patients in non-critical care setting: an endocrine society clinical practice guideline. *J Clin Endocrinol Metab*, 97(1):16–38, 2012.

[260] Guillermo E Umpierrez, Scott D Isaacs, Niloofar Bazargan, Xiangdong You, Leonard M Thaler, and Abbas E Kitabchi. Hyperglycemia: an independent marker of in-hospital mortality in patients with undiagnosed diabetes. *The Journal of Clinical Endocrinology & Metabolism*, 87(3):978–982, 2002.

[261] G van den Berghe, P Wouters, F Weekers, C Verwaest, F Bruyninckx, M Schetz, D Vlasselaers, P Ferdinande, P Lauwers, and R Bouillon. Intensive insulin therapy in critically ill patients. *N Engl J Med*, 345(19):1359–1367, 2001.

[262] Moshe Y Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for concurrency*, pages 238–266. Springer, 1996.

[263] Ramanarayan Vasudevan, Victor Shia, Yiqi Gao, Ricardo Cervera-Navarro, Ruzena Bajcsy, and Francesco Borrelli. Safe semi-autonomous control with enhanced driver modeling. In *American Control Conference (ACC), 2012*, pages 2896–2903. IEEE, 2012.

[264] ME Wallymahmed, S Dawes, G Clarke, S Saunders, N Younis, and IA MacFarlane. Hospital in-patients with diabetes: increasing prevalence and management problems. *Diabetic Medicine*, 22(1):107–109, 2005.

[265] M. Webster, C. Dixon, M. Fisher, M. Salem, J. Saunders, K.L. Koay, K. Dautenhahn, and J. Saez-Pons. Toward reliable autonomous robotic assistants through formal verification: A case study. *Human-Machine Systems, IEEE Transactions on*, PP(99):1–11, 2015.

[266] Matt Webster, Clare Dixon, Michael Fisher, Maha Salem, Joe Saunders, Kheng Koay, and Kerstin Dautenhahn. Formal verification of an autonomous personal

robotic assistant. *Formal Verification and Modeling in Human-Machine Systems*, 2014.

[267] J Weimer, S A Ahmadi, J Araujo, and others. Active Actuator Fault Detection and Diagnostics in HVAC Systems. In *Proceedings of the Fourth Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 107–114, 2012.

[268] J Weimer, S Kar, and K H Johansson. Distributed Detection and Isolation of Topology Attacks in Power Networks. In *International Conference on High Confidence Networked Systems*, pages 65–72, 2012.

[269] J Weimer, D Varagnolo, and K H Johansson. Distributed Model-invariant Detection of Unknown Inputs in Networked Systems. In *International Conference on High Confidence Networked Systems*, pages 127–134, 2013.

[270] James Weimer, Sanjian Chen (corresponding & co-first author), Amy Peleckis, Michael R. Rickels, and Insup Lee. Physiology-invariant meal detection for type 1 diabetes. *Diabetes Technology & Therapeutics*, 2016 (Accepted).

[271] James Weimer, Radoslav Ivanov, Alexander Roederer, Sanjian Chen, and Insup Lee. Parameter-invariant design of medical alarms. *IEEE Design & Test*, 32(5):9, 2015.

[272] Richard L Weinstein, Sherwyn L Schwartz, Ronald L Brazg, Jolyon R Bugler, Thomas A Peyser, and Geoffrey V McGarraugh. Accuracy of the 5-day freestyle navigator continuous glucose monitoring system comparison with frequent laboratory reference measurements. *Diabetes Care*, 30(5):1125–1130, 2007.

[273] Christopher D Wickens. *Engineering psychology and human performance* . HarperCollins Publishers, 1992.

[274] Christopher D Wickens, Huiyang Li, Amy Santamaria, Angelia Sebok, and Nadine B Sarter. Stages and levels of automation: An integrated meta-analysis. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 54, pages 389–393. SAGE Publications, 2010.

[275] EARL L WIENER and RENWICK E CURRY. Flight-deck automation: promises and problems. *Ergonomics*, 23(10):995–1011, October 2007.

[276] EL Wiener. Complacency: Is the term useful for air safety. In *Proceedings of the 26th Corporate Aviation Safety Seminar*, volume 117, 1981.

[277] Malgorzata E Wilinska, Ludovic J Chassin, Carlo L Acerini, Janet M Allen, David B Dunger, and Roman Hovorka. Simulation environment to evaluate closed-loop insulin delivery systems in type 1 diabetes. *Journal of diabetes science and technology*, 4(1):132–144, 2010.

[278] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[279] TM Wolever and Claudia Bolognesi. Prediction of glucose and insulin responses of normal subjects after consuming mixed meals varying in energy, protein, fat, carbohydrate and glycemic index. *The Journal of nutrition*, 126(11):2807–2812, 1996.

[280] David D Woods, James Tittle, Magnus Feil, and Axel Roesler. Envisioning human-robot coordination in future operations. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(2):210–218, 2004.

[281] Jiaquan Xu, Kenneth D Kochanek, Sherry L Murphy, and Betzaida Tejada-Vera. Deaths: final data for 2007. *National vital statistics reports: from the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System*, 58(19):1–19, 2010.

[282] Jun Yang, Shihua Li, Changyin Sun, and Lei Guo. Nonlinear-disturbance-observer-based robust flight control for airbreathing hypersonic vehicles. *Aerospace and Electronic Systems, IEEE Transactions on*, 49(2):1263–1275, 2013.

[283] YEON J Yang, JANG H Youn, and RICHARD N Bergman. Modified protocols improve insulin sensitivity estimation using the minimal model. *American Journal of Physiology-Endocrinology And Metabolism*, 253(6):E595–E602, 1987.

[284] Wim Zeiler, RV HOUTEN, Gert Boxem, Derrek Vissers, and Rik Maaijen. Indoor air quality and thermal comfort strategies: the human-in-the-loop approach. In *Proc. Int. Conf. Enhanced Building Oper*, 2011.

[285] Howard Zisser, Lauren Robinson, Wendy Bevier, Eyal Dassau, Christian Ellingsen, Francis J Doyle III, and Lois Jovanovic. Bolus calculator: a review of four smart insulin pumps. *Diabetes technology & therapeutics*, 10(6):441–444, 2008.